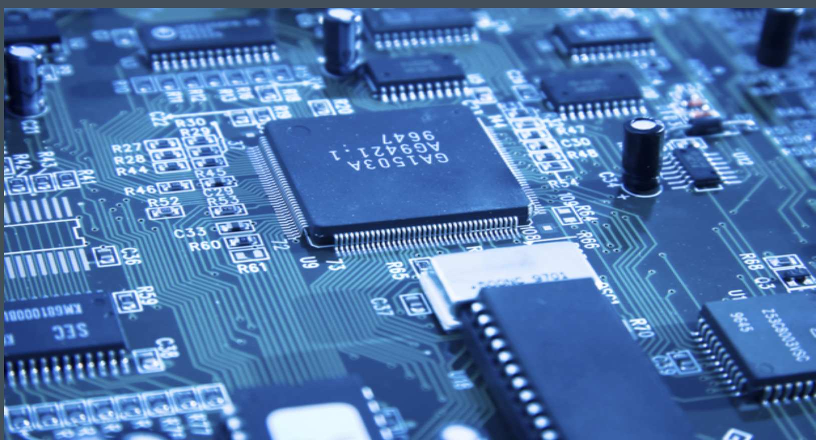
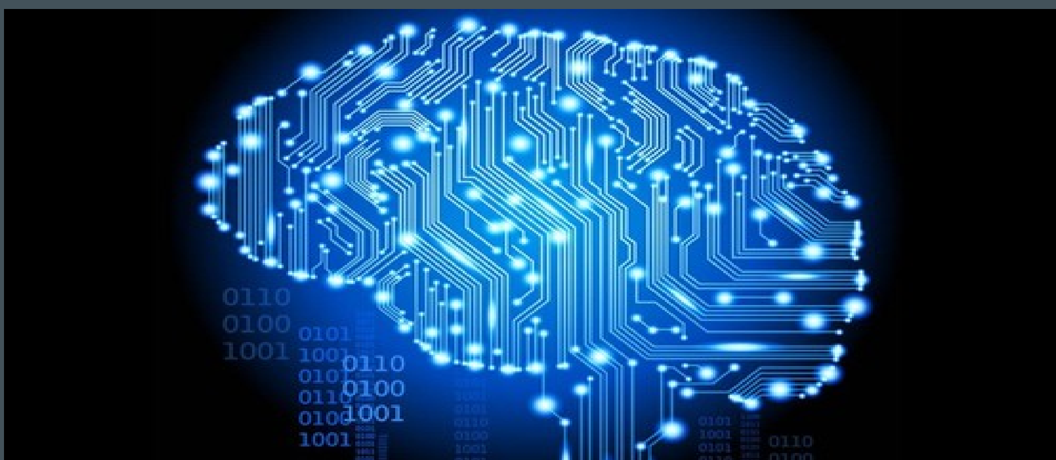
УНИВЕРЗИТЕТ У НОВОМ САДУ
ПОЉОПРИВРЕДНИ ФАКУЛТЕТ

PRIMENJENA INFORMATIKA

ZBIRKA ZADATAKA

Dr Tihomir Zoranović





2

Dr Tihomir Zoranović

PRIMENJENA INFORMATIKA

Zbirka zadataka

UNIVERZITET U NOVOM SADU
POLJOPRIVREDNI FAKULTET

Novi Sad, 2016.

EDICIJA POMOĆNI UDŽBENIK

Osnivač i izdavač edicije

Poljoprivredni fakultet, Novi Sad
Trg Dositeja Obradovića 8, 21000 Novi Sad

Godina osnivanja
1954

Glavni i odgovorni urednik edicije
Dr Nedeljko Tica*, redovni profesor*
Dekan Poljoprivrednog fakulteta

CIP - Каталогизација у публикацији
Библиотека Матице српске, Нови Сад

338.43:330(075.8)(076)

Зорановић Тихомир
Primenjena Informatika: praktikum / Tihomir Zoranović. - Novi Sad : Poljoprivredni fakultet, 2016. - 55 str. : 24 cm. - (Edicija Pomoćni udžbenik)

Tiraž 20. - Bibliografija.

ISBN 978-86-7520-356-0

a) Информатика - Практикуми
COBISS.SR-ID 301050119

Autor

Dr Tihomir Zoranović

docent

Glavni i odgovorni urednik

Dr Nedeljko Tica*, redovni profesor*

Dekan Poljoprivrednog fakulteta u Novom Sadu

Urednik

Dr Vladislav Zekić, *vanredni profesor*

Direktor Departmana za ekonomiku poljoprivrede

i sociologiju sela, Poljoprivredni fakultet u Novom Sadu

Recenzenti

Dr Bojan Srđević*, profesor Emeritus*

Poljoprivredni fakultet, Novi Sad

Dr Ivana Berković, *redovni profesor*

Tehnički fakultet, Zrenjanin

Izdavač

Univerzitet u Novom Sadu, Poljoprivredni fakultet, Novi Sad

Štampanje odobrio: Komisija za izdavačku delatnost,

Poljoprivredni fakultet, Novi Sad

Tiraž: 20

Mesto i godina štampanja: Novi Sad, 2016.

PREDGOVOR

Zbirka zadataka je namenjena studentima svih studijskih programa Poljoprivrednog fakulteta u Novom Sadu koji imaju izborni predmet Primenjena informatika. Pisana je u skladu sa akreditovanim Nastavnim planom i programom za predmet Primenjena informatika, sa ciljem da studentima omogući uspešno praćenje rada na predavanjima i vežbama.

Zbirka zadataka je koncipirana tako da pomogne studentima u savladavanju predispitnih i ispitnih obaveza u specifičnoj oblasti sa kojom se do sada nisu sretali. Detaljno i postupno su obrađene teme koje su blisko vezane za Informatiku i informacione sisteme. Prvo poglavlje obrađuje vrednovanje količine informacija kada se desi jedan ili više događaja.

Drugo poglavlje uvodi studente u svet brojnih sistema. Obrađena su tri najvažnija brojna sistema koja se koriste na svim digitalnom uređajima.

Treće poglavlje ima za cilj da ukaže na potrebu za analitičkim mišljenjem i kroz primere pokazuje kako se jednostavni problemi mogu podeliti na korake i predstaviti univerzalnim simbolima.

Četvrto poglavlje obrađuje optimizacione probleme kroz detaljne primere dve metode. Poznavanje optimizacionih metoda i pronalaženje najboljeg rešenja je danas cilj svakog donosioca odluka.

Zahvaljujem se recenzentima Prof. dr Ivani Berković a posebno Prof. dr emeritus Bojanu Srđeviću, profesoru sa dvadesetogodišnjim iskustvom na predmetu Informatika, na izuzetno korisnim sugestijama koje su doprinele da zbirka bude kompletnija.

Novi Sad, 2016. Autor

Sadržaj

1. Merenje količine informacija

Podatak je činjenica o nekom događaju ili pojavi. Ne ulazeći u istinitost tvrdnje „Danas je lep dan“ može se zaključiti da je to jedan podatak.

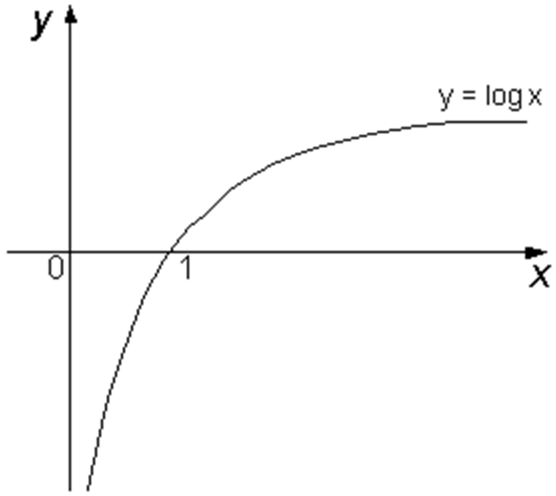
Informacija je obrađena činjenica koja ima neku svoju vrednost za osobu kojoj je činjenica izrečena. Da bi rečenica „Danas je lep dan“ postala informacija, mora da ima značaj (vrednost) za primaoca, odnosno ako primaoc putuje ili ide na plažu onda primljeni podatak obradom pretvara u informaciju.

1.1. Jedan događaj

Za meru neodređenosti nekog događaja koristi se pojam **entropija**. Klod Šenon se smatra „ocem“ teorije informacija i za entropiju se koristi njegova formula

$$H(x) = -\sum_{i=1}^{n} p_i \log_2 p_i$$

Entropija je uvek pozitivan broj jer je verovatnoća p_i manja od 1 pa je vrednost logaritamske funkcije negativna.



Slika br. 1: Logaritamska funkcija

Zbir verovatnoća događaja je jednaka jedinici, odnosno

$$\sum_{i=1}^{n} p_i = 1$$

Jedinica za entropiju je **bit**.

Ako su verovatnoće događaja jednake i ako su mogući odgovori na pitanja „Da“ i „Ne“ (ili „Tačno“ i „Netačno“) onda Šenonov obrazac ima oblik

$$H(x) = \log_2 n$$

Zadatak 1: Koliko bitova informacije sadrži podatak da je bacanjem novčića pala „glava“?

Rešenje: Pretpostavka je da se radi o idealnom novčiću, odnosno o jednako verovatnim ishodima bacanja, odnosno verovatnoća da padne „glava“ je p_{glava}=0.5 i p_{pismo}=0.5. Novčić ne može pasti na stranu.

Primenom Šenonovog obrasca količina informacija za taj događaj je

$$H(x) = \log_2 n = \log_2 2 = 1$$

Dakle, informacija o ishodu bacanja novčića nosi 1bit informacija. To odgovara i odgovoru na pitanje „koliko pitanja treba da se postavi da bi se dobio odgovor na koju stranu je pao novčić?“. Odgovor je „Jedno pitanje“. Pitanje bi moglo biti „Da li je pala glava?“ i ako je odgovor „Da“ onda se zna ishod bacanja (glava). Ako je odgovor „Ne“, opet se zna ishod bacanja (pismo). Suština je da je potrebno jedno pitanje da se sazna ishod bacanja novčića.

Zadatak 2: Koliko je pitanja potrebno postaviti da bi se utvrdilo na koji kolosek dolazi voz ako stanica ima 11 perona?

Rešenje: Primenom Šenonovog obrasca

$$H(x) = \log_2 n = \log_2 11 = 3{,}46 \approx 4$$

Pošto je nemoguće postaviti 3,46 pitanja uvek je rezultat veći ceo broj, u ovom slučaju 4 bita.

Zadatak 3: Koliko je pitanja potrebno postaviti da bi se utvrdilo koja je karta izvučena iz špila od 52 karte?

Rešenje: Primenom Šenonovog obrasca

$$H(x) = \log_2 n = \log_2 52 = 5{,}71 \approx 6$$

Pošto je nemoguće postaviti 5,71 pitanje, rezultat je 6 bitova.

1.2. Više nezavisnih događaja

Verovatnoća da će se desiti ishodi više nezavisnih događaja je jednaka proizvodu verovatnoća pojedinačnih ishoda. U slučaju bacanja novčića i kocke verovatnoće bi bile

$$p(NovčićKocka) = p(Novčić) * p(Kocka) = \frac{1}{2} * \frac{1}{6} = \frac{1}{12}$$

Bacanjem novčića i kocke moguće je dobiti 12 različitih ishoda.

Količina informacije za više nezavisnih događaja se izračunava kao zbir količina informacija pojedinačnih događaja i za bacanje novčića i kocke je

$$I(Novčić, Kocka) = \log_2 Novčić + \log_2 Kocka = \log_2 2 + \log_2 6 = 1 + 2.58 \approx 4$$

Dakle, ako se želi izračunati količina informacije za više nezavisnih događaja koristi se **Hartlijeva teorema** koja izražava maksimalnu količinu informacija za više nezavisnih događaja

$$I_{max}(n_1, n_2, \dots, n_m) = \log_2 n_1 + \log_2 n_2 + \cdots + \log_2 n_m$$

Podrazumeva se da se radi o idealnom novčiću i idealnoj kocki, odnosno da su pri bacanju novčića ishodi da je novčić pao na jednu ili drugu stranu jednako verovatni te da isto važi i za bilo koju stranu kocke ili bilo koju kartu iz špila karata.

Zadatak 1: Koliko bitova informacije nosi podatak da je bacanjem dva novčića na oba pala „glava“?

Rešenje: Primenom Hartlijeve teoreme

$$I(Novčić1, Novčić2) = \log_2 Novčić1 + \log_2 Novčić2 = \log_2 2 + \log_2 2 = 1 + 1 = 2$$

To znači da su dovoljna dva pitanja da bi se saznao ishod bacanja dva novčića. Pitanja bi mogla biti „Da li je palo pismo?“. Ako je odgovor „Da“ onda znamo ishod prvog događaja a ako je odgovor „Ne“ – takođe znamo ishod bacanja. isto pitanje se ponavlja i za drugi novčić i koristimo istu logiku. Jasno je da se radi o nezavisnim događajima pa su pitanja postavljena za jedan pa za drugi događaj.

Zadatak 2: Koliko pitanja je potrebno postaviti da bi se saznao ishod bacanja dve kocke?

Rešenje: Primenom Hartlijeve teoreme

$$I(Koc1, Koc2) = \log_2 Koc1 + \log_2 Koc2 = \log_2 6 + \log_2 6 = 2.58 + 2.58 = 5{,}16 \approx 6$$

Zadatak 3: Koliko pitanja je potrebno postaviti da bi se saznao ishod bacanja dve kocke i jednog novčića?

Rešenje: Primenom Hartlijeve teoreme

$$\begin{aligned} I(K1, K2, N) &= \log_2 K1 + \log_2 K2 + \log_2 N = \\ &= \log_2 6 + \log_2 6 + \log_2 6 = 2.58 + 2.58 + 1 = 6{,}16 \approx 7 \end{aligned}$$

1.3. Zadaci za vežbanje

Zadatak 1: Koliko je pitanja potrebno postaviti da bi se utvrdilo koji je broj pao pri bacanju kocke?

Zadatak 2: Koliko bitova nosi informacija da je iz špila „mađarica“ (32 karte) izvučena dama herc?

Zadatak 3: Koliko je bitova potrebno da bi se saopštio ishod bacanja novčića i dve kockice?

Zadatak 4: Koliko je pitanja potrebno postaviti da bi se saznao ishod bacanja novčića, jedne kockice i izvlačenja jedne karte iz špila od 52 karte?

Zadatak 5: Koliko bitova informacija nosi podatak da je na prvoj kockici pala dvojka, na drugoj trojka a na trećoj četvorka?

2. Brojni sistemi

Brojni sistemi se mogu podeliti na pozicione i nepozicione.

Karakteristika pozicionih brojnih sistema je da se vrednost broja dobija kao zbir koji se dobija kada se svaka cifra pomnoži sa osnovom brojnog sistema na stepen pozicije, odnosno kao

$$broj = \sum_{i=-m}^{n} a_i * r^i$$

odnosno

$$broj = a_n * r^n + a_{n-1} * r^{n-1} + \cdots + a_1 * r^1 + a_0 * r^0 + a_{-1} * r^{-1} + \cdots + a_m * r^m$$

U dekadnom brojnom sistemu vrednost broja 876,54 je

$876{,}54 = 8 * 10^2 + 7 * 10^1 + 6 * 10^0 + 5 * 10^{-1} + 4 * 10^{-2}$

Vrlo je važno uočiti poziciju cifre pa je tako

Pozicija	2	1	0		-1	-2
Broj	8	7	6	,	5	6

U dekadnom brojnom sistemu nulta pozicija se zove „jedinice" (tj. 10^0), prva „desetice" (10^1), druga pozicija je „stotine" (tj. 10^2) itd. Za druge brojne sisteme ne postoje takvi adekvatni nazivi.

Pošto osnova brojnog sistema može biti različita, da bi se znalo u kom brojnom sistemu se računa osnova brojnog sistema se piše u indeksu u zagradi (broj 25 se piše $25_{(10)}$).

Vrednost broja kod nepozicionih brojnih sistema se određuje na drugačiji način. Poznati rimski brojni sistem je predstavnik nepozicionih sistema koji je i danas u upotrebi. Svaka od sedam cifara ima svoju vrednost pa je

I – Cifra za broj 1

V - Cifra za broj 5

X - Cifra za broj 10

L - Cifra za broj 50

C - Cifra za broj 100

D - Cifra za broj 500

M - Cifra za broj 1000

Pravila za tumačenje vrednosti broja su:

- Cifre rimskih brojeva (**I**, **V**, **X**, **L**, **C**, **D** i **M**) imaju uvek istu vrednost nezavisno na kom mestu u broju se nalaze.
- Rimski brojevi se uvek pišu (i čitaju) s leva na desno, od najveće do najmanje cifre.
- Ako su cifre napisane tako da desna nije veća od leve, onda se vrednosti cifri sabiraju.
- Ako je vrednost leve cifre manja od desne, onda se vrednost leve cifre oduzima od vrednosti desne.
- Cifre **I**, **X** i **C** smeju se uzastopce zapisati najviše tri puta.
- Cifre **V**, **L** i **D** smeju se zapisati samo jednom.
- Cifra **I** se oduzima samo od **V** i **X**, cifra **X** samo od **L** i **C** a cifra **C** samo od **D** i **M**.

2.1. Binarni brojni sistem

Binarni brojni sistem je pozicioni brojni sistem koji ima osnovu brojnog sistema r=2 a raspolaže ciframa 0 i 1.

2.1.1. Pretvaranje dekadnog broja u binarni ekvivalent

Dekadni broj se deli sa osnovom brojnog sistema (u ovom slučaju sa 2) i rezultat deljenja se piše ispod dekadnog broja a ostatak pri deljenju sa strane. Dobijeni binarni broj se čita odole na gore.

Zadatak 1: Dekadni broj 361 pretvoriti u binarni ekvivalent.

Rešenje:

361	1
180	0
90	0
45	1
22	0
11	1
5	1
2	0
1	1
0	

Dakle, broj $361_{(10)}=101101001_{(2)}$

Kako se to može proveriti?

Pošto je binarni brojni sistem pozicioni, za njega važi da se vrednost broja izračunava kao

$$broj = \sum_{i=-m}^{n} a_i * r^i$$

pa je

$\text{broj} = 1 * 2^8 + 0 * 2^7 + 1 * 2^6 + 1 * 2^5 + 0 * 2^4 + 1 * 2^3 + 0 * 2^2 + 0 * 2^1 + 1 * 2^0 =$

$= 256 + 0 + 64 + 32 + 0 + 8 + 0 + 0 + 1 = 361_{(10)}$

Zadatak 2: Dekadni broj 1589 pretvoriti u binarni ekvivalent.

1589	1
794	0
397	1
198	0
99	1
49	1
24	0
12	0
6	0
3	1
1	1
0	

Dakle, broj $1589_{(10)}=11000110101_{(2)}$

Pretvaranje binarnog broja u dekadni ekvivalent:

$11000110101_{(2)} = 1*2^{10}+1*2^9+0*2^8+0*2^7+0*2^6+1*2^5+1*2^4+0*2^3+1*2^2+0*2^1+1*2^0 =$

$= 1024 + 512 + 0 + 0 + 0 +32 + 16 + 0 + 4 + 0 + 1 = 1589_{(10)}$

Zadatak 3: Dekadni broj 3035 pretvoriti u binarni ekvivalent.

Rešenje:

3035	1
1517	1
758	0
379	1
189	1
94	0
47	1
23	1
11	1
5	1
2	0
1	1
0	

Dakle, broj $3035_{(10)}=101111011011_{(2)}$

Pretvaranje binarnog broja u dekadni ekvivalent:

$$101111011011_{(2)} = 1*2^{11} + 0*2^{10} + 1*2^9 + 1*2^8 + 1*2^7 + 1*2^6 + 0*2^5 +$$
$$+ 1*2^4 + 1*2^3 + 0*2^2 + 1*2^1 + 1*2^0 =$$
$$= 2048 + 0 + 512 + 256 + 128 + 64 + 0 + 16 + 8 + 0 + 2 + 1 = 3035_{(10)}$$

2.1.2. Sabiranje binarnih brojeva

Sabiranje binarnih brojeva podleže pravilima sabiranja za pozicione brojne sisteme sa posebnim akcentom na raspoložive cifre (0 i 1). Tako je

Binarni	Računa se kao	Dekadni
1		1
+0		+0
1	$=1*2^0 =$	1
+1		+1
10	$=0*2^0 + 1*2^1 =$	2
+1		+1
11	$=1*2^0 + 1*2^1 =$	3
+1		+1
100	$=1*2^2 =$	4

Zadatak 1: Sabrati binarne brojeve 11010 i 10011 u binarnom brojnom sistemu i rezultat proveriti u dekadnom brojnom sistemu.

Rešenje:

$$\begin{array}{r} 11010 \\ \underline{+10011} \end{array}$$

Sabiranje počinje najmanje značajnom mestom, odnosno krajnje desnom kolonom (nulta kolona), gde je $0_{(2)}+1_{(2)}=1_{(2)}$, piše se 1, odnosno

$$\begin{array}{r} 11010 \\ \underline{+10011} \\ 1 \end{array}$$

sledeća levo (prva) kolona se računa kao $1_{(2)}+1_{(2)}=10_{(2)}$ a treba sabrati još i prenesen broj 1 pa je rezultat $1_{(2)}+1_{(2)}=10_{(2)}$. Nula se piše a 1 se prenosi na veće težinsko mesto, odnosno

$$\begin{array}{r} 11010 \\ \underline{+10011} \\ 01 \end{array}$$

druga kolona se računa kao $0_{(2)}+0_{(2)}=0_{(2)}$ a treba dodati još i prenešen broj 1 pa je $0_{(2)}+1_{(2)}=1_{(2)}$. Jedan se piše, odnosno

$$\begin{array}{r} 11010 \\ \underline{+10011} \\ 101 \end{array}$$

treća kolona se računa kao $1_{(2)}+0_{(2)}=1_{(2)}$. Jedan se piše, odnosno

```
 11010
+10011
  1101
```

četvrta kolona se računa kao $1_{(2)}+1_{(2)}=10_{(2)}$. Nula se piše a 1 se prenosi na veće težinsko mesto, odnosno

```
 11010
+10011
 01101
```

pošto u petoj koloni nema brojeva upisuje se prenesena jedinica pa je rezultat

```
 11010
+10011
101101
```

provera u dekadnom brojnom sistemu podrazumeva pretvaranje binarnih brojeva u dekadne i sabiranje u dekadnom.

broj1 $_{(10)} = 1*2^4+1*2^3+0*2^2+1*2^1+0*2^0 = 16 + 8 + 0 + 2 + 0 = 26_{(10)}$

broj2 $_{(10)} = 1*2^4+0*2^3+0*2^2+1*2^1+1*2^0 = 16 + 0 + 0 + 2 + 1 = 19_{(10)}$

broj3 $_{(10)} = 1*2^5 + 0*2^4+1*2^3+1*2^2+0*2^1+1*2^0 = 32 + 0 + 8 + 4 + 0 + 1 = 45_{(10)}$

Sve je dobro izračunato jer je $26_{(10)} + 19_{(10)} =45_{(10)}$

Zadatak 2: Sabrati binarne brojeve 11011 i 10111 u binarnom brojnom sistemu i rezultat proveriti u dekadnom brojnom sistemu.

Rešenje:

```
 11011
+10111
```

Sabiranje počinje najmanje značajnom mestom, odnosno krajnje desnom kolonom (nulta kolona), gde je $1_{(2)}+1_{(2)}=10_{(2)}$, piše se 0 a prenosi se 1 na veće značajno mesto, odnosno

```
 11011
+10111
     0
```

sledeća levo (prva) kolona se računa kao $1_{(2)}+1_{(2)}=10_{(2)}$ a treba sabrati još i prenesen broj 1 pa je rezultat $10_{(2)}+1_{(2)}=11_{(2)}$. Jedan se piše a 1 se prenosi na veće težinsko mesto, odnosno

```
 11011
+10111
    10
```

druga kolona se računa kao $0_{(2)}+1_{(2)}=1_{(2)}$ a treba dodati još i prenešen broj 1 pa je $1_{(2)}+1_{(2)}=10_{(2)}$. Nula se piše a 1 se prenosi na veće težinsko mesto, odnosno

```
 11011
+10111
   010
```

treća kolona se računa kao $1_{(2)}+0_{(2)}=1_{(2)}$ a treba dodati još i prenešen broj 1 pa je $1_{(2)}+1_{(2)}=10_{(2)}$. Nula se piše a 1 se prenosi na veće težinsko mesto, odnosno

$$\begin{array}{r} 11011 \\ +10111 \\ \hline 0010 \end{array}$$

četvrta kolona se računa kao $1_{(2)}+1_{(2)}=10_{(2)}$ a treba dodati još i prenešen broj 1 pa je $10_{(2)}+1_{(2)}=11_{(2)}$. Jedan se piše a 1 se prenosi na veće težinsko mesto, odnosno

$$\begin{array}{r} 11011 \\ +10111 \\ \hline 10010 \end{array}$$

pošto u petoj koloni nema brojeva upisuje se prenesena jedinica pa je rezultat

$$\begin{array}{r} 11011 \\ +10111 \\ \hline 110010 \end{array}$$

provera u dekadnom brojnom sistemu podrazumeva pretvaranje binarnih brojeva u dekadne i sabiranje u dekadnom.

broj1 $_{(10)} = 1*2^4+1*2^3+0*2^2+1*2^1+1*2^0 = 16 + 8 + 0 + 2 + 1 = 27_{(10)}$

broj2 $_{(10)} = 1*2^4+0*2^3+1*2^2+1*2^1+1*2^0 = 16 + 0 + 4 + 2 + 1 = 23_{(10)}$

broj3 $_{(10)} = 1*2^5 + 1*2^4+0*2^3+0*2^2+1*2^1+0*2^0 = 32 + 16 + 0 + 0 + 2 + 0 = 50_{(10)}$

Jasno je da je sve dobro izračunato jer je $27_{(10)} + 23_{(10)} = 50_{(10)}$

2.1.3. Oduzimanje binarnih brojeva

Pravila za oduzimanje u binarnom brojnom sistemu su identična kao i za sve pozicione brojne sisteme.

Zadatak 1: Od binarnog broja 1101 oduzeti binarni broj 11 i računanje proveriti u dekadnom brojnom sistemu.

Rešenje:

$$\begin{array}{r} 1101 \\ -\quad 11 \\ \hline \end{array}$$

U nultoj koloni se od 1 oduzima 1 i dobija 0.

$$\begin{array}{r} 1101 \\ -\quad 11 \\ \hline 0 \end{array}$$

U prvoj koloni je potrebno oduzeti od 0 jedinice što ne može. Potrebno je pozajmiti jedinicu sa većeg težinskog mesta pa računati 10-1 i dobija se rezultat 1.

$$\begin{array}{r} 1101 \\ -\quad 11 \\ \hline 10 \end{array}$$

U drugoj koloni potrebno je od 1 oduzeti nulu ali i vratiti pozajmljenu jedinicu tj dodati pozajmljenu jedinicu u broj koji se oduzima i to je u stvari 1-1=0

```
  1101
-   11
   010
```

U trećoj koloni treba od 1 oduzeti 0 i rezultat je 1.

```
  1101
-   11
  1010
```

provera u dekadnom brojnom sistemu podrazumeva pretvaranje binarnih brojeva u dekadne i oduzimanje u dekadnom.

broj1 $_{(10)} = 1*2^3+1*2^2+0*2^1+1*2^0 = 8 + 4 + 0 + 1 = 13_{(10)}$

broj2 $_{(10)} = 1*2^1+1*2^0 = 2 + 1 = 3_{(10)}$

broj3 $_{(10)} = 1*2^3+0*2^2+1*2^1+0*2^0 = 8 + 0 + 2 + 0 = 10_{(10)}$

Rezultat je ispravan jer je $13_{(10)} + 3_{(10)} = 10_{(10)}$

Zadatak 2: Od binarnog broja 11011 oduzeti binarni broj 101 i računanje proveriti u dekadno brojnom sistemu.

```
  11011
-   101
```

U nultoj koloni se od 1 oduzima 1 i dobija 0.

```
  11011
-   101
      0
```

U prvoj koloni je potrebno oduzeti od 1 nulu i dobija se rezultat 1.

```
  11011
-   101
     10
```

U drugoj koloni potrebno je od 0 oduzeti 1 ali to ne može. Mora se pozajmiti 1 sa većeg težinskog mesta i broj koji se oduzima je u stvari 10-1=1

```
  11011
-   101
    110
```

U trećoj koloni treba od 1 oduzeti 0 ali i vratiti pozajmljenu jedinicu pa se zapravo oduzima 1-1 i rezultat je 1.

```
  11011
-   101
   0110
```

U četvrtoj koloni treba od 1 oduzeti 0 i rezultat je 1.

```
  11011
-   101
  10110
```

provera u dekadnom brojnom sistemu podrazumeva pretvaranje binarnih brojeva u dekadne i oduzimanje u dekadnom.

broj1 $_{(10)} = 1*2^4 + 1*2^3+0*2^2+1*2^1+1*2^0 = 16 + 8 + 0 + 2 + 1 = 27_{(10)}$

broj2 $_{(10)} = 1*2^2 + 0*2^1+1*2^0 = 4 + 0 + 1 = 5_{(10)}$

broj3 $_{(10)} = 1*2^4 + 0*2^3+1*2^2+1*2^1+0*2^0 = 16 + 0 + 4 + 2 + 0 = 22_{(10)}$

Rezultat je ispravan jer je $27_{(10)} - 5_{(10)} = 22_{(10)}$

2.1.4. Množenje binarnih brojeva

Pravila za množenje u binarnom brojnom sistemu su identična kao i za sve pozicione brojne sisteme.

Zadatak 1: Pomnožiti binarne brojeve 1101 i 11 a računanje proveriti u dekadnom brojnom sistemu.

Rešenje: Posle množenja najmanje značajnom cifrom dobija se

```
  1101* 11
 ---------
  1101
```

Množenjem druge cifre množioca dobija se

```
  1101* 11
 ---------
  1101
 1101
```

Potrebno je sabrati dva broja i sabiranjem dobija se

```
  1101* 11
 ---------
  1101
 1101
-----------
100111
```

Treba uočiti da sabiranje zadnje tri kolone (nulta, prva i druga) ne predstavlja problem (1+0=1, 0+1=1, 1+0=1) ali 1+1=10. Nula se piše a 1 se prenosi na veće težinsko mesto. Četvrta kolona se računa kao 1+0=1+ preneta jedinica=10 koje se tako i piše.

Provera u dekadnom brojnom sistemu je

$1101_{(2)}=1*2^3+1*2^2+0*2^1+1*2^0=8+4+0+1=13$

$11_{(2)}= 1*2^1+1*2^0=2+1=3$

$100111_{(2)}= 1*2^5+0*2^4+0*2^3+1*2^2+1*2^1+1*2^0=32+0+0+4+2+1=39$

Množenje u dekadnom sistemu je

$13_{(10)}*3_{(10)}=39_{(10)}$

Zadatak 2: Pomnožiti binarne brojeve 11011 i 1011 a računanje proveriti u dekadnom brojnom sistemu.

Rešenje: Posle množenja najmanje i pravilnog potpisivanja dobija se

$$
\begin{array}{r}
\underline{\quad 11011 * 1011} \\
11011 \\
11011 \\
00000 \\
\underline{11011} \\
100101001
\end{array}
$$

Sabiranje u nultoj koloni je 1+0+0+0=1

Sabiranje u prvoj koloni je 1+1+0+0=10, nula se piše a 1 se prenosi u sledeću kolonu

Sabiranje u drugoj koloni je 0+1+0+0+prenetih 1 =10, nula se piše a 1 se prenosi u sledeću kolonu.

Sabiranje u trećoj koloni je 1+0+0+1+prenetih 1=11, jedan se piše a 1 se prenosi u sledeću kolonu.

Sabiranje u četvrtoj koloni je 1+1+0+1+prenetih 1=100, nula se piše a 10 se prenosi u sledeću kolonu.

Sabiranje u petoj koloni je 0+1+0+0+prenetih 10=11, jedan se piše a 1 se prenosi u sledeću kolonu.

Sabiranje u šestoj koloni je 0+0+0+1+prenetih 1=10, nula se piše a 1 se prenosi u sledeću kolonu.

Sabiranje u sedmoj koloni je 0+0+0+1+prenetih 1=10, piše se nula a se prenosi u sledeću kolonu.

Sabiranje u osmoj koloni je 0+0+0+0+prenetih 1=1, piše se jedan.

Provera u dekadnom brojnom sistemu:

$11011_{(2)}= 1*2^4 +1*2^3+0*2^2+1*2^1+1*2^0=16+8+0+2+1=27$

$1011_{(2)}= 1*2^3+0*2^2+1*2^1+1*2^0=8+0+2+1=11$

$100101001_{(2)}=1*2^8+0+2^7+0*2^6+1*2^5+0*2^4 +1*2^3+0*2^2+0*2^1+1*2^0=$

$=256+0+0+32+0+8+0+0+1=297_{(10)}$

2.1.5. Deljenje binarnih brojeva

Zadatak 1: Podeliti binarne brojeve 11110 i 101 u binarnom brojnom sistemu i rezultat proveriti u dekadnom brojnom sistemu.

Rešenje: Iz brojioca se izdvaja onoliko cifara koliko može da se podeli sa imeniocem, u ovom slučaju 111. Tako je 111:101=1 sa ostatkom koji se piše ispod.

$$
\begin{array}{l}
\underline{11110 : 101}=1 \\
\underline{101} \\
\;10
\end{array}
$$

Spušta se 1 i deli se 101:101=1 a ostatak (u ovom slučaju 0) se piše ispod

```
11110 : 101=11
101
 101
 101
   0
```

Spušta se 0 i deli se 0:101=0 a ostatak (u ovom slučaju 0) se piše ispod

```
11110 : 101=110
101
 101
 101
   00
    0
    0
```

Provera u dekadnom brojnom sistemu:

$11110_{(2)}= 1*2^4 +1*2^3+1*2^2+1*2^1+0*2^0=16+8+4+2=30$

$101_{(2)}= 1*2^2+0*2^1+1*2^0=4+0+1=5$

$110_{(2)}= 1*2^2+1*2^1+0*2^0=4+2+0=6$

Rezultat deljenja dekadnih brojeva 30:5=6 što znači da je računanje ispravno.

Zadatak 2: Podeliti binarne brojeve 11101 i 100 u binarnom brojnom sistemu i rezultat proveriti u dekadnom brojnom sistemu.

Rešenje: Iz brojioca se izdvaja onoliko cifara koliko može da se podeli sa imeniocem, u ovom slučaju 111. Tako je 111:100=1 sa ostatkom koji se piše ispod.

```
11101 : 100=1
100
 11
```

Spušta se 0 i deli se 110:100=1 a ostatak (u ovom slučaju 10) se piše ispod

```
11101 : 100=11
100
 110
 100
  10
```

Spušta se 1 i deli se 101:100=1 a ostatak (u ovom slučaju 1) se piše ispod

```
11101 : 100=111
100
 110
 100
  101
  100
    1
```

Pošto više nema brojeva, stavlja se zarez, spušta se 0 i deli se 10:100=0 a ostatak (u ovom slučaju 10) se piše ispod

11101 : 100=111,0
100
 110
 100
  101
  100
    10
     0
    10

Spušta se 0 i deli se 100:100=0 a ostatak (u ovom slučaju 0) se piše ispod

11101 : 100=111,01
100
 110
 100
  101
  100
    10
     0
    100
    100
      0

Provera u dekadnom brojnom sistemu:

$11101_{(2)}= 1*2^4 +1*2^3+1*2^2+0*2^1+1*2^0=16+8+4+1=29$

$100_{(2)}= 1*2^2+0*2^1+0*2^0=4+0+0=4$

$111,01_{(2)}= 1*2^2+1*2^1+1*2^0+0*2^{-1}+1*2^{-2}=4+2+1+0+1/4=7,25_{(10)}$

Rezultat deljenja dekadnih brojeva 29:4=7,25 što znači da je računanje ispravno.

2.2. Oktalni brojni sistem

Oktalni brojni sistem je pozicioni brojni sistem koji ima osnovu brojnog sistema r=8 a raspolaže ciframa 0, 1, 2, 3, 4, 5, 6, 7.

Pretvaranje dekadnog broja u oktalni ekvivalent

Dekadni broj se deli sa osnovom brojnog sistema (u ovom slučaju sa 8) i rezultat deljenja se piše ispod dekadnog broja a ostatak deljenja sa strane. Dobijeni oktalni broj se čita odole na gore.

Zadatak 1: Dekadni broj 314 pretvoriti u oktalni ekvivalent i proveriti rezultat.

Rešenje:

314	2
39	7
4	4
0	

Suprotni smer tj. pretvaranje oktalnog broja u dekadni ekvivalent koristi činjenicu da je oktalni brojni sistem pozicioni pa važi

broj = $4 * 8^2 + 7 * 8^1 + 4 * 8^0 = 4 * 64 + 7 * 8 + 2 * 1 = 256 + 56 + 2 = 314_{(10)}$

Zadatak 2: Dekadni broj 1375 pretvoriti u oktalni ekvivalent i proveriti rezultat.

Rešenje:

$$\begin{array}{r|l} 1375 & 7 \\ 171 & 3 \\ 21 & 5 \\ 2 & 2 \\ 0 & \end{array}$$

Suprotni smer tj pretvaranje oktalnog broja u dekadni ekvivalent koristi činjenicu da je oktalni brojni sistem pozicioni pa važi

broj = $2 * 8^3 + 5 * 8^2 + 3 * 8^1 + 7 * 8^0 = 2 * 512 + 5 * 64 + 3 * 8 + 7 * 1 =$

$= 1024 + 320 + 24 + 7 = 1375_{(10)}$

2.2.1. Sabiranje oktalnih brojeva

Sabiranje se računa na isti način kao i u svakom drugom pozicionom sistemu ali se mora voditi računa da je najveća cifra 7 (Tabela 1).

Tabela 1: Sabiranje u oktalnom brojnom sistemu

+	**1**	**2**	**3**	**4**	**5**	**6**	**7**
1	2	3	4	5	6	7	10
2	3	4	5	6	7	10	11
3	4	5	6	7	10	11	12
4	5	6	7	10	11	12	13
5	6	7	10	11	12	13	14
6	7	10	11	12	13	14	15
7	10	11	12	13	14	15	16

Zadatak 1: Sabrati oktalne brojeve 562 i 123 i rezultat proveriti u dekadnom brojnom sistemu.

Rešenje:

$$\begin{array}{r} 562 \\ + \quad 123 \\ \hline 5 \end{array}$$

Prema tablici sabiranja, nulta kolona je 3+2=5. Prva kolona se računa kao 6+2=10, nula se piše a 1 se prenosi na veće težinsko mesto.

$$\begin{array}{r} 562 \\ +\ \ 123 \\ \hline 05 \end{array}$$

Druga kolona se računa kao 5+1+prenešena jedinica=7

$$\begin{array}{r} 562 \\ +\ \ 123 \\ \hline 705 \end{array}$$

Provera u dekadnom brojnom sistemu je:

$562_{(8)} = 5 * 8^2 + 6 * 8^1 + 2 * 8^0 = 5 * 64 + 6 * 8 + 2 * 1 = 320 + 48 + 2 = 370_{(10)}$

$123_{(8)} = 1 * 8^2 + 2 * 8^1 + 3 * 8^0 = 1 * 64 + 2 * 8 + 3 * 1 = 64 + 16 + 3 = 83_{(10)}$

$705_{(8)} = 7 * 8^2 + 0 * 8^1 + 5 * 8^0 = 7 * 64 + 0 * 8 + 5 * 1 = 448 + 0 + 5 = 453_{(10)}$

Pošto je $370_{(10)} + 83_{(10)} = 453_{(10)}$ računanje u oktalnom brojnom sistemu je ispravno.

Zadatak 2: Sabrati oktalne brojeve 546 i 473 i rezultat proveriti u dekadnom brojnom sistemu.

Rešenje:

$$\begin{array}{r} 546 \\ +\ \ 473 \\ \hline 1 \end{array}$$

Prema tablici sabiranja, nulta kolona je 6+3=11. Piše se 1 i prenosi se 1. Prva kolona se računa kao 4 + 7 + prenešenih 1 = 14, četiri se piše a 1 se prenosi na veće težinsko mesto.

$$\begin{array}{r} 546 \\ +\ \ 473 \\ \hline 41 \end{array}$$

Druga kolona se računa kao 5+4+prenešena jedinica = 12. Dva se piše a 1 prenosi na veće težinsko mesto.

$$\begin{array}{r} 546 \\ +\ \ 473 \\ \hline 241 \end{array}$$

treća kolona ima samo prenešenu jedinicu pa je rezultat

$$\begin{array}{r} 546 \\ +\ \ 473 \\ \hline 1241 \end{array}$$

Provera u dekadnom brojnom sistemu je:

$546_{(8)} = 5 * 8^2 + 4 * 8^1 + 6 * 8^0 = 5 * 64 + 4 * 8 + 6 * 1 = 320 + 32 + 6 = 358_{(10)}$

$473_{(8)} = 4 * 8^2 + 7 * 8^1 + 3 * 8^0 = 4 * 64 + 7 * 8 + 3 * 1 = 256 + 56 + 3 = 315_{(10)}$

$1241_{(8)} = 1 * 8^3 + 2 * 8^2 + 4 * 8^1 + 1 * 8^0 = 1 * 256 + 2 * 64 + 4 * 8 + 1 * 1 =$

$= 512 + 128 + 32 + 1 = 673_{(10)}$

Pošto je $358_{(10)} + 315_{(10)} = 673_{(10)}$ računanje u oktalnom brojnom sistemu je ispravno.

2.2.2. Množenje oktalnih brojeva

Množenje oktalnih brojeva se može računati na dva načina. prvi je da se koristi tabela množenja u oktalnom brojnom sistemu (Tabela 2)

Tabela 2: Množenje u oktalnom brojnom sistemu

*	**1**	**2**	**3**	**4**	**5**	**6**	**7**
1	1	2	3	4	5	6	7
2	2	4	6	10	12	14	16
3	3	6	11	14	17	22	25
4	4	10	14	20	24	30	34
5	5	12	17	24	31	36	43
6	6	14	22	30	36	44	52
7	7	16	25	34	43	52	61

Drugi način je da se množi kao da se radi o dekadnom brojnom sistemu a onda da se dekadni rezultat pretvara u oktalni ekvivalent.

Zadatak 1: Pomnožiti oktalne brojeve 56 i 67 a rezultat proveriti u dekadnom brojnom sistemu.

Rešenje: Ako se pomnože cifre 6 i 7 kao dekadni brojevi dobija se dekadni broj 42. Njega treba pretvoriti u oktalni ekvivalent na poznati način

$$\begin{array}{r|l} 42 & 2 \\ 5 & 5 \\ 0 & \end{array}$$

pa je $42_{(10)} = 52_{(8)}$. Piše se 2 a 5 prenosi na veće težinsko mesto pa je

$$\begin{array}{l} \underline{56 * 67} \\ \;\; 2 \end{array}$$

Množenje cifara 5 i 7 kao dekadnih se svodi na $5*7=35_{(10)}$ što je

$$\begin{array}{r|l} 35 & 3 \\ 4 & 4 \\ 0 & \end{array}$$

pa se sledeća cifra računa kao 3 + prenesenih 5 = 8. Dekadni broj 8 je

$$\begin{array}{r|l} 8 & 0 \\ 1 & 1 \\ 0 & \end{array}$$

pa se piše 0 a prenosi se još 1, pored 4, na veće težinsko mesto. Tako je

56 * 67
02

Ukupno se prenosi 4+1=5 pa je rezultat množenja

56 * 67
502

Množenje 56 * 6 počinje sa dekadnim množenjem 6*6=36. Oktalni ekvivalent je

36	4
4	4
0	

pa se piše 4 a 4 prenosi na veće težinsko mesto

56 * 67
502
4

Sledi 5*6=30+ prenetih 4 = $34_{(10)}$. Oktalni ekvivalent je

34	2
4	4
0	

pa je

56 * 67
502
424

Sabiranje je

56 * 67
502
424
4742

Provera u dekadnom brojnom sistemu koristi činjenicu da je oktalni brojni sistem pozicioni

$56_{(8)} = 5 * 8^1 + 6 * 8^0 = 46_{(10)}$

$67_{(8)} = 6 * 8^1 + 7 * 8^0 = 55_{(10)}$

$4742_{(8)} = 4 * 8^3 + 7 * 8^2 + 4 * 8^1 + 2 * 8^0 = 2530_{(10)}$

Pošto je $46_{(10)} * 55_{(10)} = 2530_{(10)}$ računanje je ispravno.

Zadatak 2: Pomnožiti oktalne brojeve 345 i 56 a rezultat proveriti u dekadnom brojnom sistemu.

Rešenje: Ako se pomnože cifre 5 i 6 kao dekadni brojevi dobija se dekadni broj 30. Njega treba pretvoriti u oktalni ekvivalent na poznati način

30	6
3	3
0	

pa je $30_{(10)} = 36_{(8)}$. Piše se 6 a 3 prenosi na veće težinsko mesto pa je

$$\begin{array}{r} \underline{345 * 56} \\ 6 \end{array}$$

Množenje cifara 4 i 6 kao dekadnih se svodi na 4*6+prenetih $3=27_{(10)}$ što je

27	3
3	3
0	

pa se piše 3 a prenosi se 3. Tako je

$$\begin{array}{r} \underline{345 * 56} \\ 36 \end{array}$$

Množenje cifara 3 i 6 kao dekadnih se svodi na 3*6+prenetih $3=21_{(10)}$ što je

21	5
2	2
0	

pa se piše 5 a prenosi se 2. Tako je

$$\begin{array}{r} \underline{345 * 56} \\ 2536 \end{array}$$

Množenje 345 * 5 počinje sa dekadnim množenjem 5*5=25. Oktalni ekvivalent je

25	1
3	3
0	

pa se piše 1 a 3 prenosi na veće težinsko mesto

$$\begin{array}{r} \underline{345 * 56} \\ 2536 \\ 1 \end{array}$$

Sledi 4*5=20+ prenetih 3 = $23_{(10)}$. Oktalni ekvivalent je

23	7
2	2
0	

pa je

$$\begin{array}{r} \underline{\quad 345 * 56} \\ 2536 \\ 71 \end{array}$$

Sledi 3*5=15 + prenetih 2 = $17_{(10)}$. Oktalni ekvivalent je

17	1
2	2
0	

pa je

$$\begin{array}{r} \underline{\quad 345 * 56} \\ 2536 \\ 2171 \end{array}$$

Sabiranje je

$$\begin{array}{r} \underline{\quad 345 * 56} \\ 2536 \\ \underline{2171} \\ 24446 \end{array}$$

Provera u dekadnom brojnom sistemu koristi činjenicu da je oktalni brojni sistem pozicioni

$345_{(8)} = 3 * 8^2 + 4 * 8^1 + 5 * 8^0 = 229_{(10)}$

$56_{(8)} = 5 * 8^1 + 6 * 8^0 = 46_{(10)}$

$24446_{(8)} = 2 * 8^4 + 4 * 8^3 + 4 * 8^2 + 4 * 8^1 + 6 * 8^0 = 10534_{(10)}$

Pošto je $229_{(10)} * 46_{(10)} = 10534_{(10)}$ računanje je ispravno.

2.3. Heksadecimalni brojni sistem

Heksadecimalni brojni sistem je pozicioni brojni sistem koji ima osnovu brojnog sistema r=16 a raspolaže ciframa 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F. Slova se koriste zbog nedovoljnog broja cifara u dekadnom brojnom sistemu (samo 10).

Pretvaranje dekadnog broja u heksadecimalni ekvivalent

Dekadni broj se deli sa osnovom brojnog sistema (u ovom slučaju sa 16) i rezultat deljenja se piše ispod dekadnog broja a ostatak deljenja sa strane. Dobijeni heksadecimalni broj se čita odole na gore.

2.3.1. Pretvaranje dekadnog broja u heksadecimalni

Zadatak 1: Pretvoriti dekadni broj 154 u heksadecimalni ekvivalent

Rešenje: Dekadni broj 154 se deli sa osnovom heksadecimalnog brojnog sistema (rezultat je 9 koje se piše ispod) a ostatak je 10. Dekadno 10 je ekvivalentno cifri A u heksadecimalnom sistemu. Sledeće deljenje je 9:16 gde je rezultat 0 (piše se ispod) a ostatak (9) piše se sa desne strane. Rezultat se čita od dole na gore pa je

154	A
9	9
0	

Provera u dekadnom brojnom sistemu:

$9A_{(16)} = 9 * 16^1 + 10 * 16^0 = 144 + 10 = 154_{(10)}$

Zadatak 2: Pretvoriti dekadni broj 2895 u heksadecimalni ekvivalent

Rešenje: Dekadni broj 2895 se deli sa osnovom heksadecimalnog brojnog sistema (rezultat je 180 koje se piše ispod) a ostatak je 15. Dekadno 15 je ekvivalentno cifri F u heksadecimalnom sistemu. Sledeće deljenje je 180:16 gde je rezultat 11 (piše se ispod) a ostatak (4) piše se sa desne strane. Deljenje 11:16 daje rezultat 0 (piše se ispod) a ostatak (11 dekadno tj B heksadecimalno) piše se sa desne strane. Rezultat se čita od dole na gore pa je

2895	F
180	4
11	B
0	

Provera u dekadnom brojnom sistemu:

$B4F_{(16)} = 11 * 16^2 + 4 * 16^1 + 15 * 16^0 = 2816 + 64 + 15 = 2895_{(10)}$

2.3.2. Sabiranje u heksadecimalnom brojnom sistemu

Sabiranje u heksadecimalnom brojnom sistemu je po istim pravilima za sve pozicione brojne sisteme a specifično zbog cifara koje su na raspolaganju (Tabela 3).

+	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**A**	**B**	**C**	**D**	**E**	**F**
1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10
2	3	4	5	6	7	8	9	A	B	C	D	E	F	10	11
3	4	5	6	7	8	9	A	B	C	D	E	F	10	11	12
4	5	6	7	8	9	A	B	C	D	E	F	10	11	12	13
5	6	7	8	9	A	B	C	D	E	F	10	11	12	13	14
6	7	8	9	A	B	C	D	E	F	10	11	12	13	14	15
7	8	9	A	B	C	D	E	F	10	11	12	13	14	15	16
8	9	A	B	C	D	E	F	10	11	12	13	14	15	16	17
9	A	B	C	D	E	F	10	11	12	13	14	15	16	17	18
A	B	C	D	E	F	10	11	12	13	14	15	16	17	18	19
B	C	D	E	F	10	11	12	13	14	15	16	17	18	19	1A
C	D	E	F	10	11	12	13	14	15	16	17	18	19	1A	1B
D	E	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C
E	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D
F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E

Tabela 3: Sabiranje u heksadecimalnom brojnom sistemu

Zadatak 1: Sabrati heksadecimalne brojeve C72 i 157 i rezultat proveriti u dekadnom brojnom sistemu.

Rešenje:

$$\begin{array}{rr} & \text{C72} \\ + & 157 \\ \hline & 9 \end{array}$$

Prema tablici sabiranja, nulta kolona je 2+7=9. Piše se 9. Prva kolona se računa kao 7 + 5 = 12 a to je heksadecimalno C pa se piše C.

$$\begin{array}{rr} & \text{C72} \\ + & 157 \\ \hline & \text{C9} \end{array}$$

Druga kolona se računa kao C+1 = D. Tako je

$$\begin{array}{rr} & \text{C72} \\ + & 157 \\ \hline & \text{DC9} \end{array}$$

Provera u dekadnom brojnom sistemu je:

$C72_{(16)}=12 * 16^2 + 7 * 16^1 + 2 * 16^0 = 12 * 256 + 7 * 16 + 2 * 1 = 3072 + 112 + 2 = 3186_{(10)}$

$157_{(16)} = 1 * 16^2 + 5 * 16^1 + 7 * 16^0 = 1 * 256 + 5 * 16 + 7 * 1 = 256 + 80 + 7 = 343_{(10)}$

$DC9_{(16)} = 13 * 16^2 + 12 * 16^1 + 9 * 16^0 = 13 * 256 + 12 * 16 + 9 * 1 =$

$= 3328 + 192 + 9 = 3529_{(10)}$

Pošto je $3186_{(10)} + 343_{(10)} = 3529_{(10)}$ računanje u heksadecimalnom brojnom sistemu je ispravno.

Zadatak 2: Sabrati heksadecimalne brojeve B7D i 1F7 i rezultat proveriti u dekadnom brojnom sistemu.

Rešenje:

$$\begin{array}{r} B7D \\ +\quad 1F7 \\ \hline 4 \end{array}$$

Prema tablici sabiranja, nulta kolona je D+7=$13_{(10)}$ + $7_{(10)}$ = $20_{(10)}$= $14_{(16)}$. Piše se 4 a 1 prenosi na veće težinsko mesto. Prva kolona se računa kao 7 + F + preneta jedinica = $7_{(10)}$ + $15_{(10)}$ + $1_{(10)}$ = $23_{(10)}$ = $17_{(16)}$. Piše se 7 a prenosi 1.

$$\begin{array}{r} B7D \\ +\quad 1F7 \\ \hline 74 \end{array}$$

Druga kolona se računa kao B+1+ prenetih 1 = D. Tako je

$$\begin{array}{r} B7D \\ +\quad 1F7 \\ \hline D74 \end{array}$$

Provera u dekadnom brojnom sistemu je:

$B7D_{(16)}=11 * 16^2 + 7 * 16^1 + 13 * 16^0 = 11 * 256 + 7 * 16 + 13 * 1 = 2816 + 112 + 13 =$
$=2941_{(10)}$

$1F7_{(16)} = 1 * 16^2 + 15 * 16^1 + 7 * 16^0 = 1 * 256 + 15 * 16 + 7 * 1 = 256 + 240 + 7 = 503_{(10)}$

$D74_{(16)} = 13 * 16^2 + 7 * 16^1 + 4 * 16^0 = 13 * 256 + 7 * 16 + 4 * 1 =$

$= 3328 + 112 + 4 = 3444_{(10)}$

Pošto je $2941_{(10)} + 503_{(10)} = 3444_{(10)}$ računanje u heksadecimalnom brojnom sistemu je ispravno.

2.3.3. Množenje heksadecimalnih brojeva

Množenje heksadecimalnih brojeva se može računati na dva načina.

Prvi je da se koristi tabela množenja u heksadecimalnom brojnom sistemu (Tabela 4)

Drugi način je da se množi kao da se radi o dekadnom brojnom sistemu a onda da se dekadni rezultat pretvara u heksadecimalni ekvivalent.

Pošto je drugi način bliži računanju koje se poznaje u primerima će se koristiti drugi način. Vrlo je bitno da se u jednom broj ne smeju upisivati cifre od različitih brojnih sistema.

Tabela 4: Množenje u heksadecimalnom brojnom sistemu

*	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
1	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
2	2	4	6	8	A	C	D	10	12	14	16	18	1A	1C	1E
3	3	6	9	C	F	21	15	18	1B	1E	21	24	27	2A	2D
4	4	8	C	10	14	18	1C	20	24	28	2C	30	34	38	3C
5	5	A	F	14	19	1E	23	28	2D	32	37	3C	41	46	4B
6	6	C	12	18	1F	24	2A	30	36	3C	42	48	4E	54	5A
7	7	E	15	1C	23	2A	31	38	3F	46	4D	54	5B	62	69
8	8	10	18	20	28	30	38	40	48	50	58	60	68	70	78
9	9	12	1B	24	20	36	3F	48	51	5A	63	6C	75	7E	87
A	A	14	1E	28	32	3C	46	50	5A	64	6E	78	82	8C	96
B	B	16	21	2C	37	42	4D	58	63	6E	79	84	8F	9A	15
C	C	18	24	30	3C	48	54	60	6C	78	84	90	9C	A8	B4
D	D	1A	27	34	41	4E	5B	68	75	82	8F	9C	A9	B6	C3
E	E	1C	2A	38	46	54	62	70	7E	8C	9A	A8	B6	C4	D2
F	F	1E	2D	3C	4B	5A	69	78	87	96	A5	B4	C3	D2	E1

Zadatak 1: Pomnožiti heksadecimalne brojeve F9 i 8D a rezultat proveriti u dekadnom brojnom sistemu.

Rešenje: Da bi se pomnožile cifre 9 i D potrebno ih je pretvoriti u dekadni ekvivalent pa se množenje svodi na množenje dekadnih brojeva 9 i 13. Kao rezultat dobija se dekadni broj 117. Njega treba pretvoriti u heksadecimalni ekvivalent na poznati način

117	5
7	7
0	

pa je $117_{(10)} = 75_{(16)}$. Piše se 5 a 7 prenosi na veće težinsko mesto pa je

$$\frac{F9 * 8D}{\;\;5}$$

Množenje cifara $F_{(16)}$ i $D_{(16)}$ kao dekadnih se svodi na 15*13+prenetih 7=$202_{(10)}$ što je

202	A
12	C
0	

pa se piše A a prenosi se C na veće težinsko mesto. Tako je

$$\frac{F9 * 8D}{CA5}$$

Množenje $F9_{(16)}$ * $8_{(16)}$ počinje sa dekadnim množenjem 9*8=72. heksadecimalni ekvivalent je

72	8
4	4
0	

pa se piše 8 a 4 prenosi na veće težinsko mesto

F9 * 8D
CA5
8

Sledi heksadecimalno množenje $F_{(16)}*8_{(16)}$ što se svodi na dekadno množenje 15*8=120+ prenetih 4 = $124_{(10)}$. Heksadecimalni ekvivalent je

124	C
7	7
0	

pa je

F9 * 8D
CA5
7C8

Sabiranje je

F9 * 8D
CA5
7C8
8925

jer se sabiranjem nulte kolone dobija 5.

Prva kolona se računa kao heksadecimalni A+8 što je dekadnih 10+8= 18 a kako je $18_{(10)} = 12_{(16)}$ piše se 2 a 1 nosi na veće težinsko mesto.

Druga kolona se računa kao $C_{(16)} + C_{(16)}$ + prenetih 1 = $12_{(10)} + 12_{(10)} + 1 = 25_{(10)} = 19_{(16)}$ pa se piše 9 a prenosi 1 na veće težinsko mesto.

Treća kolona se računa kao $7_{(16)} + 1_{(16)} = 8_{(16)}$.

Provera u dekadnom brojnom sistemu koristi činjenicu da je heksadecimalni brojni sistem pozicioni

$F9_{(16)} = 15 * 16^1 + 8 * 16^0 = 249_{(10)}$

$8D_{(16)} = 8 * 16^1 + 13 * 16^0 = 141_{(10)}$

$8925_{(16)} = 8 * 16^3 + 9 * 16^2 + 2 * 16^1 + 5 * 16^0 = 35109_{(10)}$

Pošto je $249_{(10)} * 141_{(10)} = 35109_{(10)}$ računanje je ispravno.

Zadatak 2: Pomnožiti heksadecimalne brojeve D5A i C9 a rezultat proveriti u dekadnom brojnom sistemu.

Rešenje: Da bi se pomnožile cifre A i 9 potrebno ih je pretvoriti u dekadni ekvivalent pa se množenje svodi na množenje dekadnih brojeva 10 i 9. Kao rezultat dobija se dekadni broj 90. Njega treba pretvoriti u heksadecimalni ekvivalent na poznati način

$$\begin{array}{r|l} 90 & A \\ 5 & 5 \\ 0 & \end{array}$$

pa je $90_{(10)} = 5A_{(16)}$. Piše se A a 5 prenosi na veće težinsko mesto pa je

$$\begin{array}{l} \underline{\text{D5A * C9}} \\ \quad\;\, \text{A} \end{array}$$

Množenje cifara $5_{(16)}$ i $9_{(16)}$ se svodi na $5_{(10)}*9_{(10)}$+prenetih $5=50_{(10)}$ što je

$$\begin{array}{r|l} 50 & 2 \\ 3 & 3 \\ 0 & \end{array}$$

pa se piše 2 a prenosi se 3 na veće težinsko mesto. Tako je

$$\begin{array}{l} \underline{\quad \text{D5A * C9}} \\ \quad\;\, \text{2A} \end{array}$$

Množenje cifara $D_{(16)}$ i $9_{(16)}$ kao dekadnih se svodi na $13_{(10)}*9_{(10)}$+prenetih $3=120_{(10)}$ što je

$$\begin{array}{r|l} 120 & 8 \\ 7 & 7 \\ 0 & \end{array}$$

pa se piše 8 a prenosi se 7 na veće težinsko mesto. Tako je

$$\begin{array}{l} \underline{\quad \text{D5A * C9}} \\ \;\; \text{782A} \end{array}$$

Množenje $D5A_{(16)}$ * $C_{(16)}$ počinje sa množenjem $A_{(16)}*C_{(16)}=10_{(10)}*12_{(10)}=120_{(10)}$. Heksadecimalni ekvivalent je

$$\begin{array}{r|l} 120 & 8 \\ 7 & 7 \\ 0 & \end{array}$$

pa se piše 8 a 7 prenosi na veće težinsko mesto

$$\begin{array}{l} \underline{\quad \text{D5A * C9}} \\ \text{782A} \\ \quad\;\, 8 \end{array}$$

Sledi heksadecimalno množenje $5_{(16)}*C_{16)}$ što se svodi na dekadno množenje $5_{(10)}*12_{(10)}=60_{(10)}$+ prenetih 7 = $67_{(10)}$. Heksadecimalni ekvivalent je

$$\begin{array}{r|l} 67 & 3 \\ 4 & 4 \\ 0 & \end{array} \uparrow$$

piše se 3 a 4 nosi na veće težinsko mesto, pa je

$$\begin{array}{r} \underline{\quad D5A * C9} \\ 782A \\ 38 \end{array}$$

Heksadecimalno množenje $D_{(16)}*C_{(16)}$ se svodi na dekadno množenje $13_{(10)}*12_{(10)}$ + prenetih 4=$160_{(10)}$. Heksadecimalni ekvivalent je

$$\begin{array}{r|l} 160 & 0 \\ 10 & A \\ 0 & \end{array} \uparrow$$

piše se 0 a A nosi na veće težinsko mesto, pa je

$$\begin{array}{r} \underline{\quad D5A * C9} \\ 782A \\ A038 \end{array}$$

Sabiranje je

$$\begin{array}{r} \underline{\quad D5A * C9} \\ 782A \\ \underline{A038} \\ A7BAA \end{array}$$

jer se sabiranjem nulte kolone dobija A.

Prva kolona se računa kao heksadecimalni $2_{(16)}+8_{(16)}$ što je dekadnih $2_{(10)}+8_{(10)}$= 10 a kako je $10_{(10)} = A_{(16)}$ piše se A.

Druga kolona se računa kao $8_{(16)} + 3_{(16)} = 8_{(10)} + 3_{(10)} = 11_{(10)} = B_{(16)}$ pa se piše B.

Treća kolona se računa kao $7_{(16)} + 0_{(16)} = 7_{(16)}$.

Četvrta kolona se računa kao $0_{(16)} + A_{(16)} = A_{(16)}$.

Provera u dekadnom brojnom sistemu koristi činjenicu da je heksadecimalni brojni sistem pozicioni

$D5A_{(16)} = 13 * 16^2 + 5 * 16^1 + 10 * 16^0 = 3418_{(10)}$

$C9_{(16)} = 12 * 16^1 + 9 * 16^0 = 201_{(10)}$

$A7BAA_{(16)} = 10 * 16^4 + 7 * 16^3 + 11 * 16^2 + 10 * 16^1 + 10 * 16^0 = 687018_{(10)}$

Pošto je $3418_{(10)} * 201_{(10)} = 687018_{(10)}$ računanje je ispravno.

2.4. Pretvaranje heksadecimalnog broja u oktalni

Izračunavanje je mogu će na dva načina:

prvi način je preko dekadnog brojnog sistema kada se heksadecimalni broj pretvori u dekadni ekvivalent a onda se dekadni broj preračuna u oktalni broj,

drugi način je preko binarnog brojnog sistema korišćenjem činjenice da je osnova binarnog brojnog sistema 2, da je osnova oktalnog brojnog sistema 3 i da je osnova heksadecimalnog brojnog sistema 4. Postupak je direktan i heksadecimalni broj se cifru po cifru direktno prevodi u binarni ekvivalent a onda se cifre dobijenog binarnog broja dele u grupo po 3 i direktno, cifru po cifru, pretvaraju u oktalni broj.

Zadatak 1: Heksadecimalni broj D946A5 pretvoriti u oktalni ekvivalent

Rešenje: Heksadecimalni broj se pretvara u binarni ekvivalent cifru po cifru

D	9	4	6	A	5
1101	1001	0100	0110	1010	0101

kada se binarni broj prikaže u grupama po 3 cifre onda se dobija

110	110	010	100	011	010	100	101
6	6	2	4	3	2	4	5

donji broj je oktalni ekvivalent početnog heksadecimalnog broja.

Zadatak 2: Oktalni broj 5744655 pretvoriti u heksadecimalni ekvivalent

Rešenje: Heksadecimalni broj se pretvara u binarni ekvivalent cifru po cifru

5	7	4	4	6	5	5
101	111	100	100	110	101	101

kada se binarni broj prikaže u grupama po 4 cifre onda se dobija

1	0111	1100	1001	1010	1101
1	7	C	9	A	D

donji broj je heksadecimalni ekvivalent početnog oktalnog broja.

2.5. Zadaci za vežbu

Zadatak 1: Pomnožiti binarne brojeve $1001011_{(2)}$ i $11011_{(2)}$ u binarnom brojnom sistemu a rezultat izraziti u dekadnom i oktalnom brojnom sistemu (rezultat: $11111101001_{(2)} = 2025_{(10)} = 3751_{(8)}$)

Zadatak 2: Pomnožiti oktalne brojeve $372_{(8)}$ i $465_{(8)}$ u oktalnom brojnom sistemu a rezultat izraziti u dekadnom i heksadecimalnom brojnom sistemu (rezultat: $226702_{(8)} = 77250_{(10)} = 12DC2_{(16)}$)

Zadatak 3: Pomnožiti heksadecimalne brojeve 3D7 i F6 u heksadecimalnom brojnom sistemu a rezultat izraziti u dekadnom i oktalnom brojnom sistemu (rezultat: $3B09A_{(16)} = 241818_{(10)} = 730232_{(8)}$)

Zadatak 4: Izračunati $10010010_{(2)} - 10111_{(2)}$ a rezultat izraziti u oktalnom i heksadecimalnom brojnom sistemu.

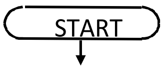
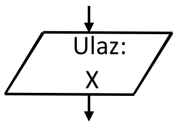
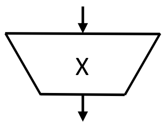
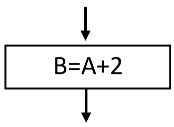
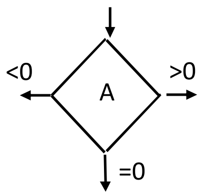
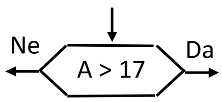
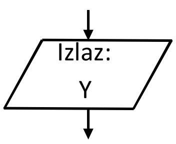
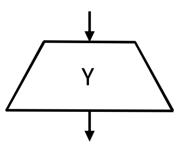
Zadatak 5: Pretvoriti oktalni broj $372465_{(8)}$ u heksadecimalni ekvivalent.

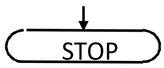
3. ALGORITMI

Algoritam je procedura, recept ili postupak za rešavanje problema istog tipa.

Algoritam se može predstaviti govornim jezikom, grafički (blok dijagramom), programskim jezikom i pseudokodom.

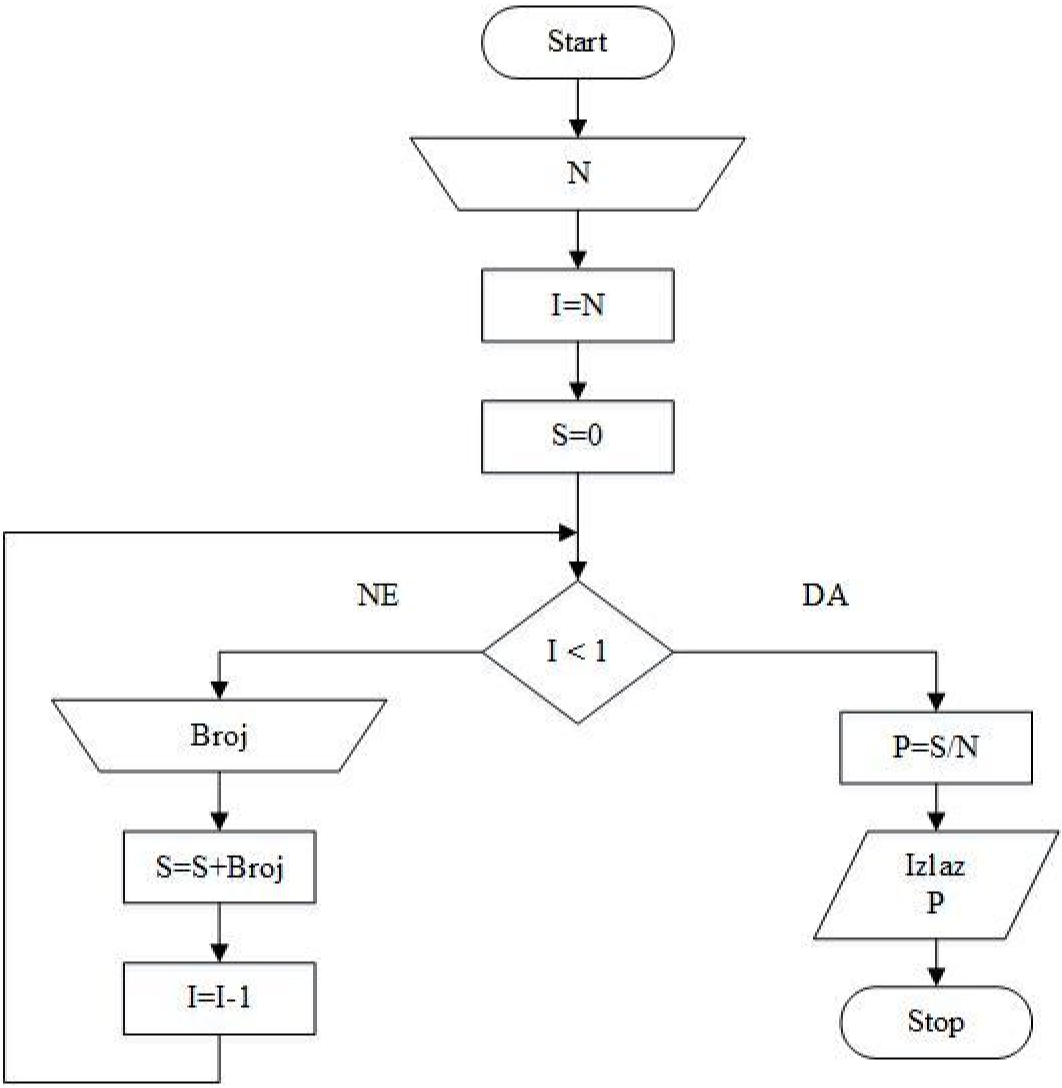
Simboli za grafički način predstavljanja koraka algoritma su standardizovani i tumače se samo na jedan način. Simboli su:

Simbol	*Značenje*
	Početak algoritma
 	Unos podataka. Uneti podatak se dodeljuje u promenljivu x
	Vrednost izraza sa desne strane se dodeljuje u promenljivu na levoj strani
	U zavisnosti od vrednosti promenljive A nastavlja se jednom od grana
	U zavisnosti od istinitosti izraza nastavlja se jednom od grana
 	Na izlazu se dobija vrednost promenljive navedene u simbolu

	Kraj algoritma

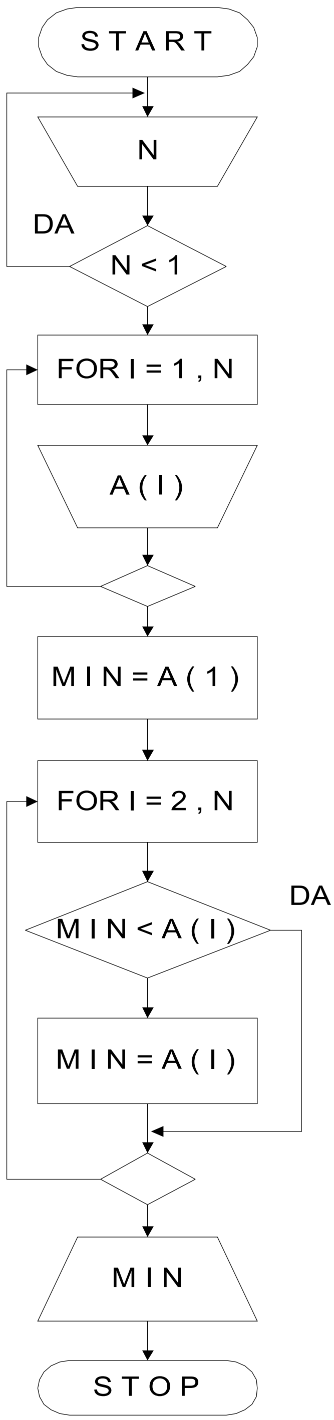
Zadatak 1: Nacrtati algoritam koji izračunava prosek od N unetih brojeva.

Rešenje: Prvo se učitava od koliko brojeva treba praviti prosek i to se dodeljuje u promenljivu N. Taj broj se prepisuje u promenljivu I koja broji koliko još brojeva treba sabrati. Početni zbir se postavlja na 0. Pita se da li je broj I manji od 1. Ako jeste promenljivoj P se dodeljuje prosek, daje se na izlaz i zaustavlja se algoritam. Ako I nije manje od 1 onda se u promenljivu Broj učitava novi broj, sabira sa dosadašnjom sumom, smanjuje se I za jedan jer je potrebno sabrati jedan broj manje i ponovo se pita da li ima još brojeva za sabiranje. Ako ima, učitava se novi broj, sabira sa dosadašnjom sumom, smanjuje promenljiva I i pita se da li još ima brojeva za sabiranje.



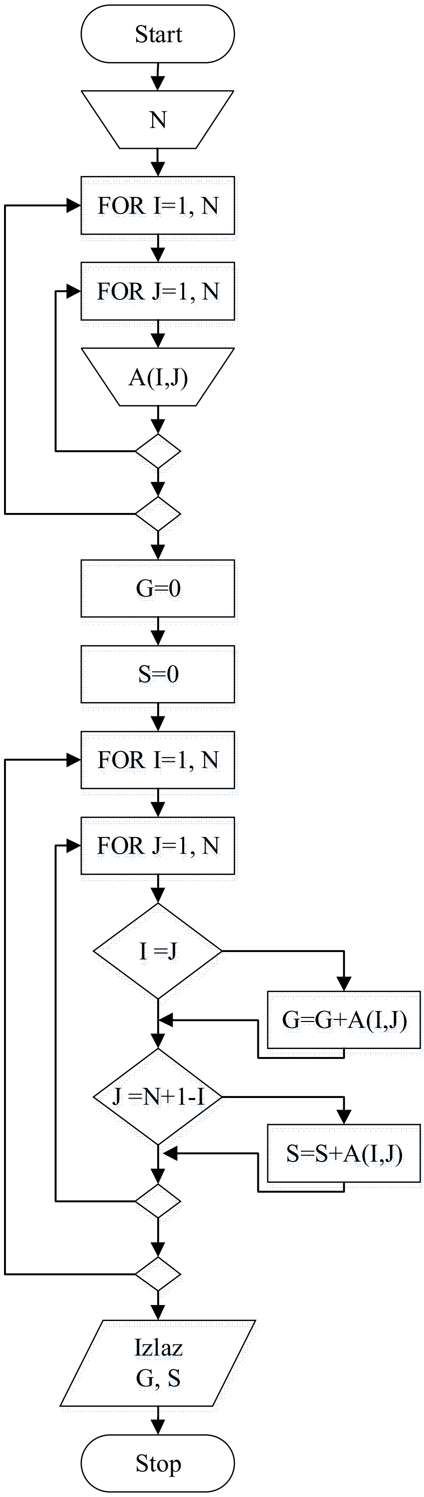
Zadatak 2: Nacrtati algoritam koji pronalazi najmanji broj od unetih brojeva.

Rešenje: Prvo se učitava broj N, odnosno koliko brojeva ukupno treba uneti. For petljom se unose brojevi u niz A (ima ih ukupno N komada pri čemu je prvi broj smešten u A(1), drugi u A(2) itd.). Prvi broj u nizu tj A(1) se proglašava za do sada najmanji broj. Nova for petlja koja ide od 2 do N omogućava da se uporede svi članovi niza od drugog do N-tog sa do sada najmanjim brojem. Ako do sada najmanji broj nije manji od nekog člana niza onda taj član niza postaje do sada najmanji broj i poredi se sa preostalim članovima niza. Kada se završi for petlja u promenljivoj MIN se nalazi najmanji broj koji se daje na izlaz.



Zadatak 3: Nacrtati algoritam koji pronalazi sumu glavne i sumu sporedne dijagonale kvadratne matrice.

Rešenje: Prvo se učitava broj N, odnosno dimenzija matrice. Zatim se unose elementi matrice i to po prvo prvi red, pa drugi itd. Elementi koji se nalaze na glavnoj dijagonali imaju poziciju gde su indeks reda i indeks kolone jednaki, odnosno sabiraju se elementi A(1,1)+A(2,2)+...+A(n,n). Elementi na sporednoj dijagonali su A(1,n)+A(2,n-1)+...A(n,1), odnosno kada indeks reda ide od 1 do n onda je indeks kolone n+1-i.



3.1. Zadaci za vežbanje

Zadatak 1: Nacrtati algoritam koji pronalazi najmanji zajednički sadržalac brojeva A i B koji se unose na ulazu.

Zadatak 2: Nacrtati algoritam koji pronalazi najveći zajednički delilac brojeva A i B koji se unose na ulazu.

Zadatak 3: Nacrtati algoritam koji pronalazi najveći palindrom od pomnožena dva trocifrena broja. Na izlazu dati najveći palindrom i trocifrene brojeve od kojih je dobijen. Pomoć: palindrom je broj koji, kad se čita cifru po cifru, ima istu vrednost ako se čita sa leva na desno i sa desna na levo (npr. 626 ili 38583).

Zadatak 4: Nacrtati algoritam koji pronalazi N-ti član Fibonačijevog niza. Pomoć: N-ti član Fibonačijevog niza je zbir prethodna dva člana (npr. prvih nekoliko članova je 1,1,2,3,5,8,... itd.).

Zadatak 5: Nacrtati algoritam koji ispisuje proste brojeve od 1 do unetog broja N. Pomoć: uneti broj N i u for petlji od 1 do N proveriti da li je broj prost, odnosno deljiv samo sa 1 i sa samim sobom.

4. OPTIMIZACIONE METODE

Optimizacija je deo matematičke oblasti Operaciona istraživanja a predstavlja postupak pronalaženja najboljeg rešenja uz poštovanje datih uslova. Optimalno rešenje se može tražiti raspoloživim metodama optimizacije (linearno programiranje, nelinearno programiranje, celobrojno programiranje, dinamičko programiranje, transportni problemi, redovi čekanja, teorija igara, višekriterijumska optimizacija itd.) a koja će se metoda primeniti zavisi od problema koji se želi optimizirati.

Imajući u vidu veliki broj raspoloživih metoda optimizacije, na sledećim stranicama će biti opisan postupak pronalaženja optimalnog rešenja linearnim programiranjem grafičkom i simpleks (analitičkom) metodom. Linearno programiranje podrazumeva da su funkcija cilja i sva ograničenja linearnog tipa. Razmatraće se samo jednokriterijumska varijanta (postoji samo jedna funkcija cilja).

4.1. Grafička metoda

Grafička metoda omogućava da se ograničenja i funkcija cilja nacrtaju i time se označava oblast dozvoljenih (mogućih) rešenja i jasno uočava tačka optimalnog rešenja. Nedostatak metode je

1. zahtev da se mora vrlo precizno i krupno crtati (teško se pročita druga ili treća decimala)
2. neupotrebljiva je za više od 2 promenljive (za tri promenljive je potrebno crtati u tri dimenzije a za četiri promenljive grafikon treba da je četvorodimenzionalan što se ne može na jednom grafikonu nacrtati).

Zadatak 1: Vulkanizer na poljoprivrednom gazdinstvu popravlja traktorske gume. Za popravku prednje gume potreban je 1 sat rada prese za gumu i 1 sat rada vulkanizera na montaži i demontaži gume. Za popravku zadnje traktorske gume potreban je 1 sat rada prese za gume i 2 sata rada vulkanizera. Presa za gume se može koristiti najviše 4 sata a vulkanizer može raditi najviše 6 sati. Za popravljenu prednju gumu ostvaruje se profit od 4 evra po komadu a za popravljenu zadnju gumu 6 evra po komadu. Optimizirati strukturu popravki tako da se ostvari najveći profit.

Rešenje: Prvo se uočavaju promenljive koje treba optimizovati a to su prednja guma (x_1) i zadnja guma (x_2). Uočavaju se ograničenja – raspoloživo vreme rada prese za gume (4 sata) i raspoloživo vreme rada vulkanizera (6 sati). Formira se nejednačina ograničenja za presu

$$1\ x_1 + 1x_2 \leq 4$$

što se tumači da je za popravku jedne prednje gume potrebno 1 čas/kom i za popravku jedne zadnje gume 1 čas/kom korišćenja prese a ukupno se presa za gume može koristiti 4 časa. Vrlo je važno proveriti da li se jedinice u nejednačini slažu, odnosno kada se napiše sa jedinicama

$$1\left[\frac{čas}{\cancel{kom}}\right]\ x_1[\cancel{kom}] + 1\left[\frac{čas}{\cancel{kom}}\right]\ x_2[\cancel{kom}]\ \leq\ 4\ [čas]$$

sve je ispravno jer se sabiraju časovi i časovi a ograničenje je u časovima.

Isto to važi i za ograničenje rada vulkanizera

$$1\left[\frac{čas}{\cancel{kom}}\right]\ x_1[\cancel{kom}] + 2\left[\frac{čas}{\cancel{kom}}\right]\ x_2[\cancel{kom}]\ \leq\ 6\ [čas]$$

Postoji još jedno ograničenje koje ne piše ali je logično, a to je da su

$x_1 \geq 0$, $x_2 \geq 0$

jer se ne može popraviti npr. -3 gume.

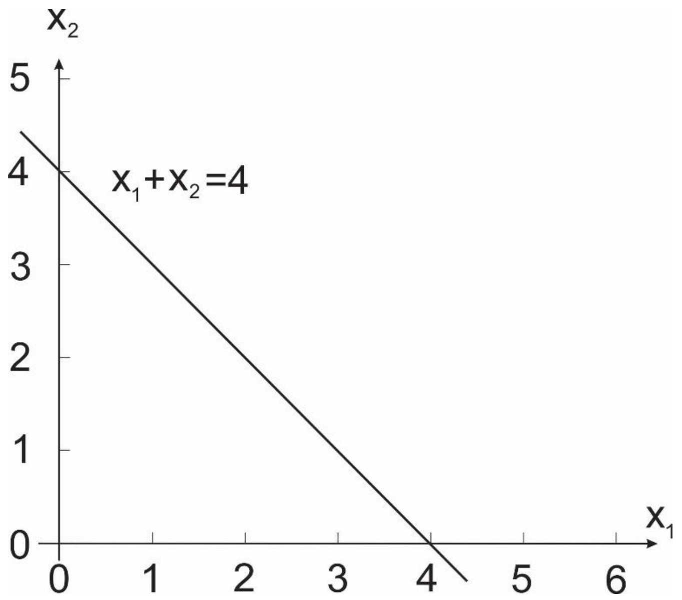
Da bi se pronašlo optimalno rešenje potrebno je da postoji cilj optimizacije. U zadatku je to optimiziranje maksimalnog profita pa je funkcija cilja

$$4\left[\frac{evra}{\cancel{kom}}\right]\ x_1[\cancel{kom}] + 6\left[\frac{evra}{\cancel{kom}}\right]\ x_2[\cancel{kom}]\ \rightarrow max\ [evra]$$

Dakle, ograničenja su:

za presu za gume	$1\ x_1 + 1x_2 \leq 4$
za vulkanizera	$1\ x_1 + 2x_2 \leq 6$
uslov nenegativnosti	$x_1 \geq 0$, $x_2 \geq 0$
funkcija cilja	$Z\!:\ 4\ x_1 + 6\ x_2\ \rightarrow max$

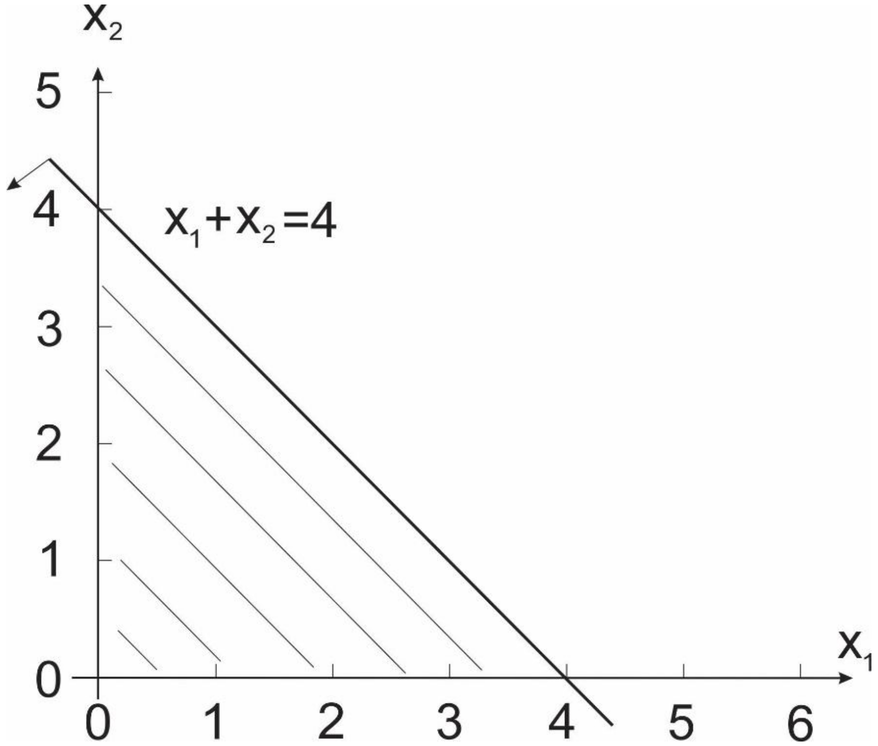
Rešavanje problema optimizacije grafičkom metodom počinje crtanjem koordinatnog sistem i to samo prvog kvadranta, odnosno u delu gde su $x_1 > 0$ i $x_2 > 0$. Na koordinatni sistem se crta prvo ograničenje (za presu za gume). Ograničenje se posmatra kao jednačina.



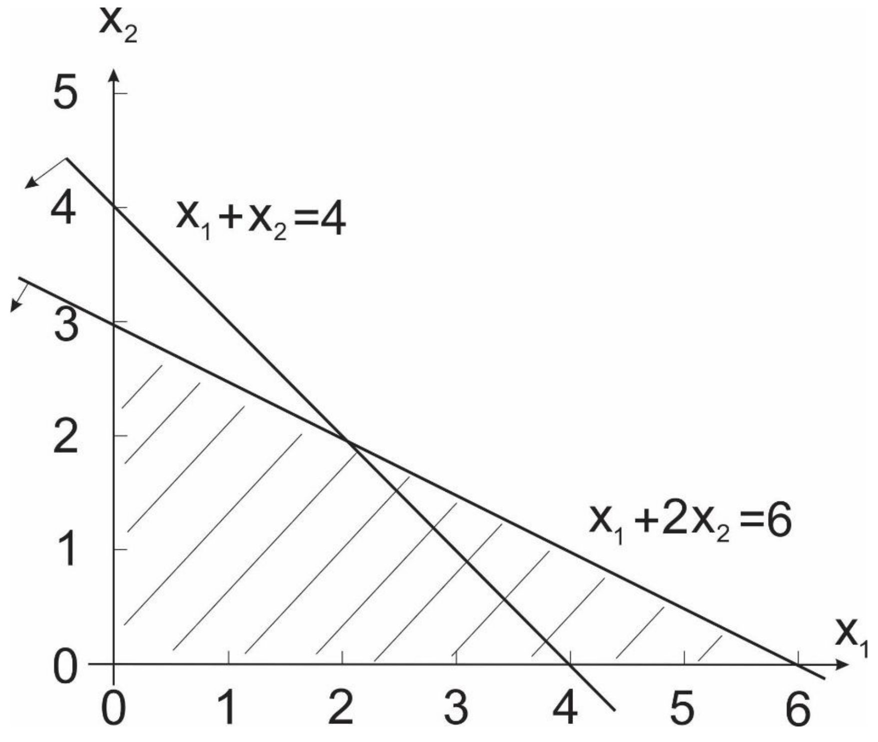
Pošto je ograničenje nejednačina, nacrtana prava je granica oblasti koja zadovoljava nejednačinu. Određivanje oblasti je najlakše ako se u nejednačinu uvrste tačke koordinatnog sistema. Ako se uvrste $x_1 = 0$ i $x_2 = 0$ u nejednačinu dobija se onda je

$$0 + 0 \leq 4$$

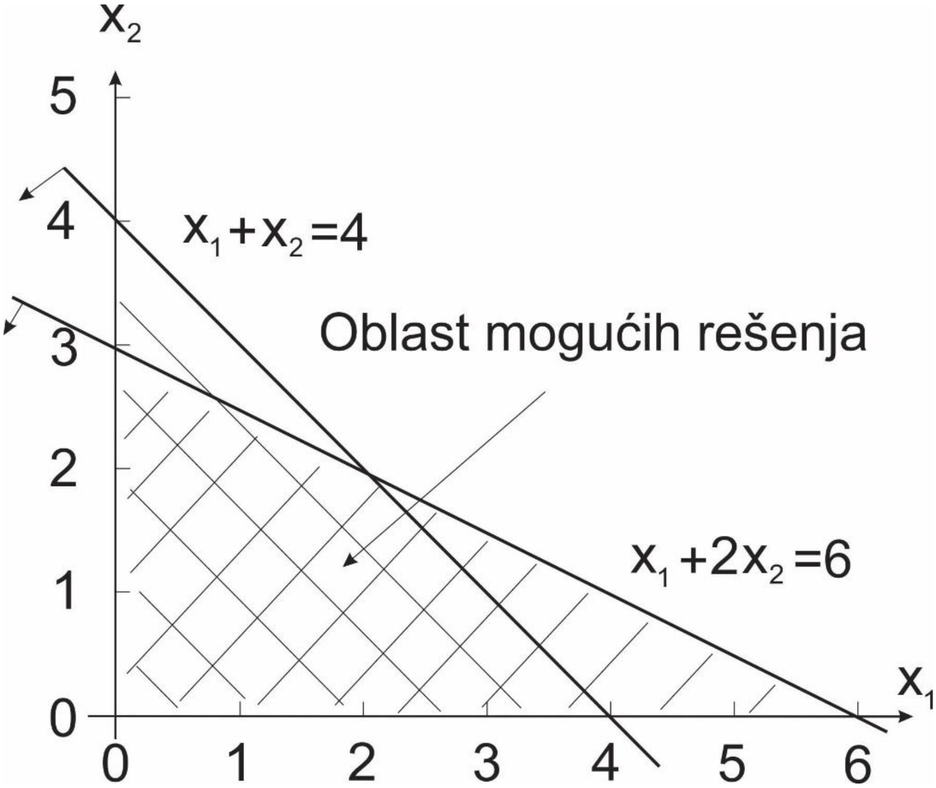
a to znači da tačka (0,0) jeste u oblasti pa je



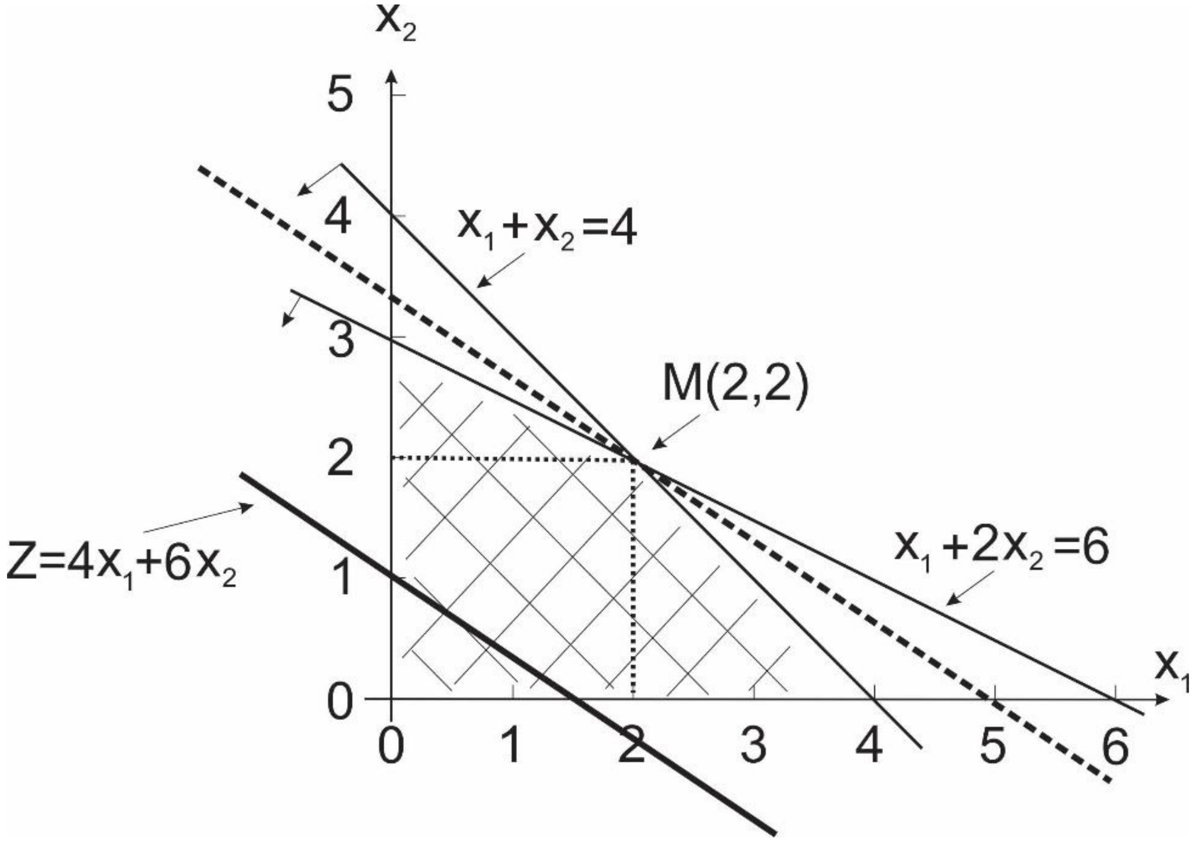
Crtanje drugog ograničenja je na sličan način, ograničenje se posmatra kao jednačina, nacrta prava a onda odredi oblast



Oba ograničenja čine oblast mogućih rešenja ili oblast dozvoljenih rešenja koja je u preseku oba ograničenja



Ostaje da se nacrta funkcija cilja i da se prava translira do najdalje tačke unutar oblasti mogućih rešenja a to je tačka M.



Da bi se pronašao optimum, potrebno je pročitati koordinate tačke M a to je

$x_1 = 2$, $x_2 = 2$. Potrebno je proveriti da li rešenje zadovoljava ograničenja, odnosno

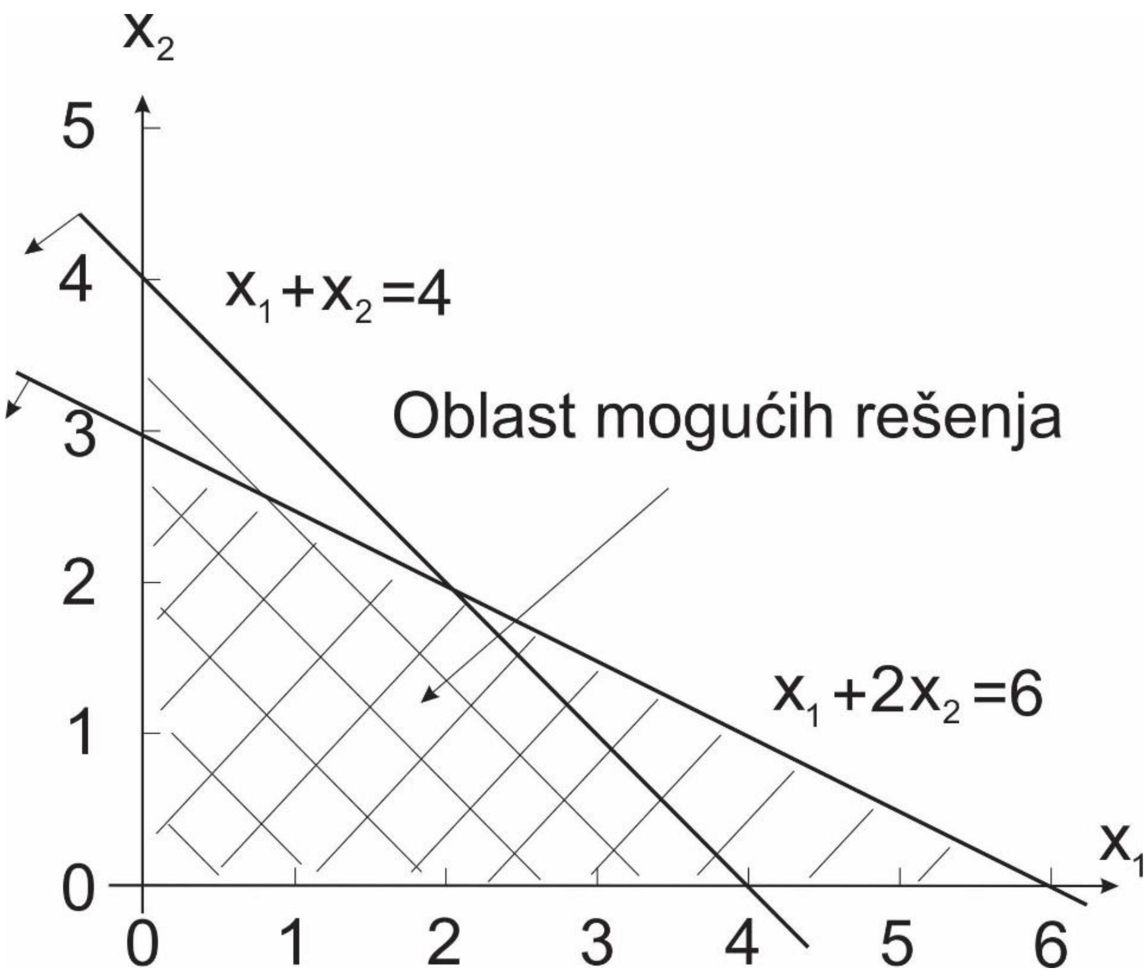
za presu za gume	$1 x_1 + 1x_2 \leq 4$	$2 + 2 \leq 4$	Zadovoljeno
za vulkanizera	$1 x_1 + 2x_2 \leq 6$	$2 + 4 \leq 6$	Zadovoljeno
uslov nenegativnosti	$x_1 \geq 0$, $x_2 \geq 0$		Zadovoljeno
funkcija cilja	$Z\colon\ 4\, x_1 + 6\, x_2\ \rightarrow max$	8 + 12 =20	Maksimum

što znači da je optimalno rešenje da se popravlja 2 prednje i 2 zadnje gume i da se može ostvariti profit od 20 evra. Metodi optimizacije garantuju da ne postoji nijedan drugi odnos koji zadovoljava ograničenja a daje veći profit.

Kakvo bi rešenje bilo kada bi funkcija cilja bila drugačija, odnosno ako je zadatak

Zadatak 2: Vulkanizer na poljoprivrednom gazdinstvu popravlja traktorske gume. Za popravku prednje gume potreban je 1 sat rada prese za gumu i 1 sat rada vulkanizera na montaži i demontaži gume. Za popravku zadnje traktorske gume potreban je 1 sat rada prese za gume i 2 sata rada vulkanizera. Presa za gume se može koristiti najviše 4 sata a vulkanizer može raditi najviše 6 sati. Za popravljenu prednju gumu ostvaruje se profit od 1 evra po komadu a za popravljenu zadnju gumu 5 evra po komadu. Optimizirati strukturu popravki tako da se ostvari najveći profit.

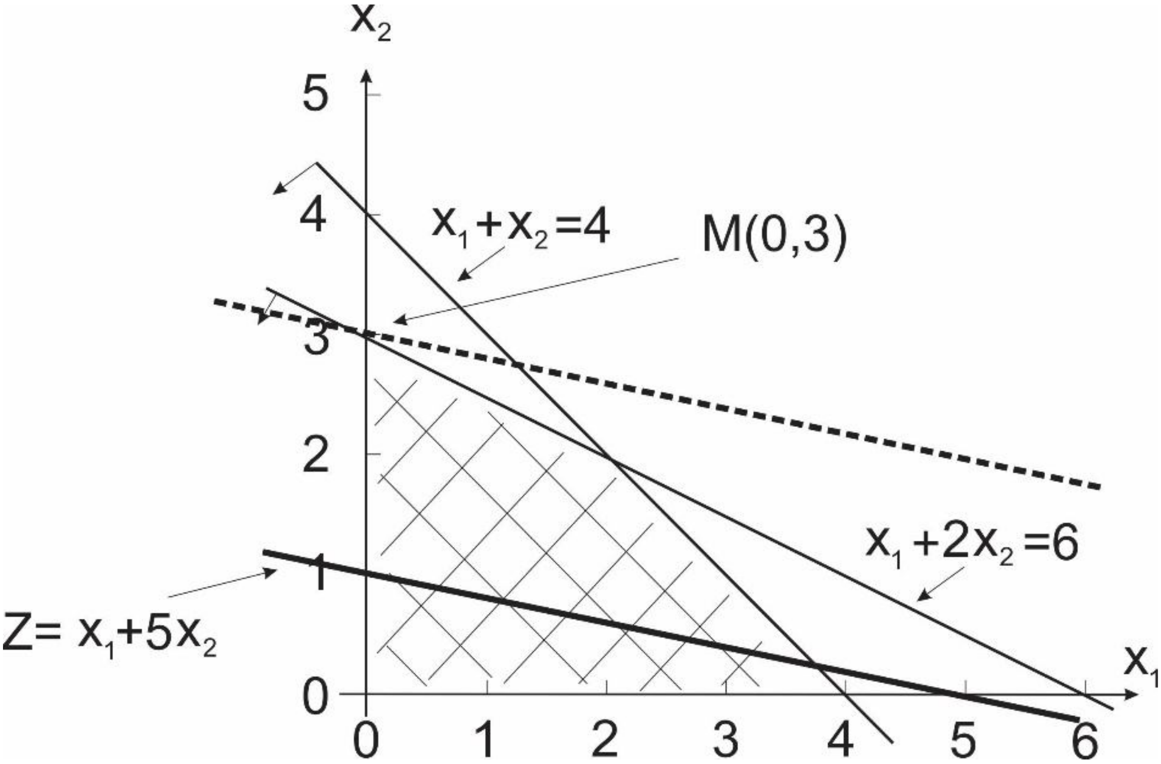
Rešenje: Razlika u odnosu na prethodni zadatak je samo u drugačijem optimizacionom zadatku. Dakle crtanje ograničenja i oblasti dozvoljenih rešenja je jednako



Potrebno je nacrtati funkciju cilja

$$Z\colon\ 1\,x_1 + 5\,x_2\ \rightarrow max$$

Crtanje funkcije cilja je najjednostavnije ako se ona posmatra kao jednačina i izjednači sa najmanjim zajedničkim sadržaocem za brojeve 5 i 1 a to je 5. Takva jednačina se lako crta i samo je potrebno tu pravu translirati do najdalje tačke unutar oblasti dozvoljenih rešenja.



Tačka optimuma M(0,3) omogućava maksimalni profit od 15 evra a optimalno rešenje je da se popravljaju samo zadnje gume.

4.2. Simpleks metoda

Simpleks metoda je optimizacioni matematički postupak koji prevazilazi nedostatke grafičkog metoda (broj promenljivih) već uvek daje optimalno rešenje. Postupak obuhvata:

- formiranje ograničenja
- definisanje funkcije cilja
- pretvaranje ograničenja i funkcije cilja u standardnu formu
- formiranje prve simpleks tabele
- izračunavanje svih ostalih simpleks tabela po pravilima.

Postupak će se objasniti na primeru.

Zadatak 1: Voćnjak prodaje oblačinsku višnju po 1000evra/t i čačanski rubin po 1500evra/t. Voće se može brati mašinom beračicom koja se može koristiti najviše 12000 časova ili se mogu uposliti radnici najviše 8000časova.

Kapaciteti su: za oblačinsku višnju, produktivnost mašina je 2časa/t a radnika 1čas/t. Za čačanski rubin, mašinom je potrebno 2časa/t a radnicima 2čas/t.

Na tržištu se može plasirati do 4000t oblačinske višnje i do 3000t višnje čačanski rubin.

U kojim količinama treba brati oblačinsku višnju i čačanski rubin da bi se ostvario maksimalni profit? Rešiti simpleks metodom.

Rešenje:

prvo se formiraju ograničenja

mašina beračica	$2x_1 + 2x_2 \leq 12000$
radnici	$1x_1 + 2x_2 \leq 8000$
tržište za oblačinsku višnju	$x_1 + \quad \leq 4000$
tržište za čačanski rubin	$x_2 \leq 3000$
uslov nenegativnosti	$x_1 \geq 0$, $x_2 \geq 0$
funkcija cilja	$Z\!: \ 1000\,x_1 + 1500\,x_2 \ \rightarrow max$

Standardna forma je omogućava da se nejednačine pretvore u jednačine i pri njenom formiranju je neophodno poštovati sledeća pravila:

ako je ograničenje $\leq$ sa leve strane se dodaje jedna slek (manjkovna) promenljiva koja preuzima neiskorišćeni resurs

ako je ograničenje = sa leve strane se dodaje jedna surplus (viškovna) promenljiva koja označava vrednost koja je dostignuta preko ograničenja.

ako je ograničenje $\geq$ sa leve strane se oduzima jedna surplus (viškovna) promenljiva i dodaje jedna veštačka (artifical) promenljiva.

U funkciji cilja slek i surplus promenljive imaju uvek koeficient 0. Ako je ciljna funkcija maksimizacija onda promenljiva artifical ima koeficient –M a ako je ciljna funkcija minimizacija koeficient uz nju je M pri čemu je M dovoljno veliki broj, teoretski beskonačno velik.

Pošto su ograničenja u zadatku $\leq$ u svaku relaciju se dodaje manjkovna promenljiva a u ciljnu funkciju uz nju ide koeficient 0.

Standardna forma:

mašina beračica	$2x_1 + 2x_2 + x_3 = 12000$
radnici	$1x_1 + 2x_2 + x_4 = 8000$
tržište za oblačinsku višnju	$x_1 + \quad + x_5 = 4000$
tržište za čačanski rubin	$x_2 + \quad + x_6 = 3000$
uslov nenegativnosti	$x_1 \geq 0$, $x_2 \geq 0$
funkcija cilja	$Z\!: \ 1000\,x_1 + 1500\,x_2 + 0\,(x_3 + x_4 + x_5 + x_6) \rightarrow max$

prva simpleks tabela se formira prepisivanjem brojeva iz standardne forme u tabelu pa je

Linija	Baza	Vrednost	x1	x2	x3	x4	x5	x6
1	x3	12000	2	2	1	0	0	0
2	x4	8000	1	2	0	1	0	0
3	x5	4000	1	0	0	0	1	0
4	x6	3000	0	1	0	0	0	1
5	/	0	1000	1500	0	0	0	0

Potrebno je pronaći pivot (radnu) kolonu a to je kolona u kojoj je najveći broj u zadnjem redu, odnosno u redu ciljne funkcije.

Linija	Baza	Vrednost	x1	x2	x3	x4	x5	x6
1	x3	12000	2	2	1	0	0	0
2	x4	8000	1	2	0	1	0	0
3	x5	4000	1	0	0	0	1	0
4	x6	3000	0	1	0	0	0	1
5	/	0	1000	1500	0	0	0	0

Pivot (radni) red je red u kome je količnik između kolone Vrednost i pivot kolone najmanji.

Za liniju 1 količnik je 12000:2=6000, za liniju 2 je 8000:2= 4000, za liniju3 4000:0= beskonačno, za liniju 4 3000:1=3000. Najmanji količnik je u liniji 4 i to je pivot (radni) red. U preseku pivot reda i pivot kolone nalazi se pivot element.

Linija	Baza	Vrednost	x1	x2	x3	x4	x5	x6
1	x3	12000	2	2	1	0	0	0
2	x4	8000	1	2	0	1	0	0
3	x5	4000	1	0	0	0	1	0
4	x6	3000	0	1	0	0	0	1
5	/	0	1000	1500	0	0	0	0

Za izračunavanje redova u sledećoj tabeli važe sledeća pravila:

- pivot red se računa tako što se svi brojevi u redu dele sa pivot elementom.
- u pivot redu u koloni baza postojeća promenljiva se zamenjuje promenljivom u vrhu pivot kolone
- ostali redovi (koji nisu pivot red) se množe koeficientom kojim se u pivot koloni dobiju nule tako da se u istoj koloni element u pivot redu pomnožen koeficientom sabira sa elementom u istoj koloni a u redu koji se računa.

Poštujući ova pravila prvi red sledeće tabele se računa na sledeći način:

koeficient je -2 jer pivot element (1) pomnožen sa -2 i sabran sa elementom u pivot koloni a 1. redu daje rezultat 0. Sada se tim koeficientom množe svi elementi u pivot redu i sabiraju sa brojevima u 1. koloni.

Linija	Baza	Vrednost	x1	x2	x3	x4	x5	x6
1	x3	12000	2	2	1	0	0	0
2	x4	8000	1	2	0	1	0	0
3	x5	4000	1	0	0	0	1	0
4	x6	3000	0	1	0	0	0	1
5	/	0	1000	1500	0	0	0	0

/(-2)/(-1500)

Linija 6 se računa na sledeći način: kolona vrednost se računa kao 3000*(-2)+12000=6000. kolona x1 se računa kao 0*(-2)+2=2, kolona x2 je 1*(-2)+2=0, kolona x3 se računa kao 0*(-2)+1=1, kolona x4 je 0*(-2)+0=0, kolona x5 je 0*(-2)+0=0, x6 je 1*(-2)+0=-2

Linija	Baza	Vrednost	x1	x2	x3	x4	x5	x6
6	x3	6000	2	0	1	0	0	-2

Da bi se dobio 7. red potrebno je 4. red pomnožiti sa (-2) i sabrati odgovarajuće pozicije. Kolona vrednost se računa kao 3000*(-2)+8000=2000, kolona x1 se računa kao 0*(-2)+1=1, kolona x2 je 1*(-2)+2=0, kolona x3 se računa kao 0*(-2)+0=0, kolona x4 je 0*(-2)+1=1, kolona x5 je 0*(-2)+0=0, x6 je 1*(-2)+0=-2

Linija	Baza	Vrednost	x1	x2	x3	x4	x5	x6
6	x3	6000	2	0	1	0	0	-2
7	x4	2000	1	0	0	1	0	-2

Pošto 8. red već ima u pivot koloni vrednost 0, može se ceo red prepisati bez izmena

Linija	Baza	Vrednost	x1	x2	x3	x4	x5	x6
6	x3	6000	2	0	1	0	0	-2
7	x4	2000	1	0	0	1	0	-2
8	x5	4000	1	0	0	0	1	0

9. red je pivot red i u kolonu Baza upisuje se promenljiva iz pivot kolone a računa se tako da se ceo red podeli sa pivot elementom. Pošto je pivot element 1 onda nema nikakve promene osim što se u bazu upisuje x2 pa je

Linija	Baza	Vrednost	x1	x2	x3	x4	x5	x6
6	x3	6000	2	0	1	0	0	-2
7	x4	2000	1	0	0	1	0	-2
8	x5	4000	1	0	0	0	1	0
9	x2	3000	0	1	0	0	0	1

10. red se računa kao i ostali redovi, koeficient je 1500 pa je u koloni vrednost 3000*(-1500)=-4 500 000, kolana x1 je 0*(-1500)+1000=1000, x2 je 1*(-1500)+1500=0, x3 je 0*(-1500)+0=0, x4 je 0*(-1500)+0=0, x5 je 0*(-1500)+0=0, x6 je 1*(-1500)+0=-1500.

Linija	Baza	Vrednost	x1	x2	x3	x4	x5	x6
6	x3	6000	2	0	1	0	0	-2
7	x4	2000	1	0	0	1	0	-2
8	x5	4000	1	0	0	0	1	0
9	x2	3000	0	1	0	0	0	1
10	/	-4 500 000	1000	0	0	0	0	-1500

Optimizacija je završena kada u zadnjem redu nema brojeva veći od 0. U tabeli postoji kolona veća od 0 i ta kolona (x1) postaje nova pivot kolona. Pivot red će biti onaj sa najmanjim koeficientom. Koeficienti su; 6000:2=3000, 2000:1=2000, 4000:1=4000 i 3000:0= beskonačno. Najmanji koeficient je u liniji 7. Količnici kojima treba množiti pivot element da bi se u koloni dobile nule su takođe prikazani.

Linija	Baza	Vrednost	x1	x2	x3	x4	x5	x6	
6	x3	6000	2	0	1	0	0	-2	
7	x4	2000	1	0	0	1	0	-2	/(-2) /(-1) /(-1000)
8	x5	4000	1	0	0	0	1	0	
9	x2	3000	0	1	0	0	0	1	
10	/	-4 500 000	1000	0	0	0	0	-1500	

Poštujući pravila računanja sledeća simpleks tabela je

Linija	Baza	Vrednost	x1	x2	x3	x4	x5	x6
11	x3	2000	0	0	1	-2	0	2
12	x1	2000	1	0	0	1	0	-2
13	x5	2000	0	0	0	-1	1	2
14	x2	3000	0	1	0	0	0	1
15	/	-6 500 000	0	0	0	-1000	0	500

Pošto i dalje u zadnjem redu promenljivih postoji broj veći od 0 neophodno je ponoviti postupak i formirati sledeću simpleks tabelu određivanjem pivot kolone, pivot reda i koeficienata

Linija	Baza	Vrednost	x1	x2	x3	x4	x5	x6	
11	x3	2000	0	0	1	-2	0	2	/+ /(-1) /(-1/2) /(-250)
12	x1	2000	1	0	0	1	0	-2	
13	x5	2000	0	0	0	-1	1	2	
14	x2	3000	0	1	0	0	0	1	
15	/	-6 500 000	0	0	0	-1000	0	500	

Izračunata simpleks tabela je

Linija	Baza	Vrednost	x1	x2	x3	x4	x5	x6
16	x6	1000	0	0	1/2	-1	0	1
17	x1	4000	1	0	1	-1	0	0
18	x5	0	0	0	-1	1	1	0
19	x2	2000	0	1	-1/2	1	0	0
20	/	-7 000 000	0	0	-250	-500	0	0

Kako u zadnjem redu u kolonama sa promenljivima nema broja većeg od 0 donosi se zaključak da je postupak završen i da je potrebno pročitati rezultat. Optimalno rešenje je

x1=4000, x2=2000 i ostvaren profit je 7 000 000 (da bi se dobio iznos optimizacije potrebno je rezultat uvek pomnožiti sa -1).

Kada se uvrste promenljive u ograničenja i funkciju cilja dobija se potvrda da je dobijeni rezultat optimalan.

Kada se ovaj problem unese u program Lingo koji je specijalizovan za probleme optimizacije dobija se sledeće rešenje

```
Global optimal solution found.
Objective value:                              7000000.
Infeasibilities:                              0.000000
Total solver iterations:                             1
Elapsed runtime seconds:                          0.03

Model Class:                                        LP

Total variables:                   2
Nonlinear variables:               0
Integer variables:                 0
```

```
Total constraints:              5
Nonlinear constraints:          0

Total nonzeros:                 8
Nonlinear nonzeros:             0


 Variable        Value        Reduced Cost
       X1     4000.000            0.000000
       X2     2000.000            0.000000

      Row    Slack or Surplus      Dual Price
        1     7000000.            1.000000
        2     0.000000            0.000000
        3     0.000000            750.0000
        4     0.000000            250.0000
        5     1000.000            0.000000
```

Zadatak 2: Poljoprivredno gazdinstvo želi da proizvodi pšenicu i kukuruz na 7 ha površine. U magacinu se nalazi 64 l pesticida i 42 mc đubriva. Da bi se ostvario profit od 6000 dinara po ha zasejane pšenice potrebno je uložiti 3mc/ha đubriva i 9 l/ha pesticida. Za profit od 9000 din/ha kukuruza potrebno je uložiti 6 mc/ha đubriva i 3 l/ha pesticida. Struktuirati setvu tako da se postigne najveći profit primenom simpleks metode.

Rešenje :

prvo se formiraju ograničenja

površina	$x_1 + \quad x_2 \leq 7$
pesticidi	$9\,x_1 + 3\,x_2 \leq 64$
đubrivo	$3\,x_1 + 6\,x_2 \leq 42$
uslov nenegativnosti	$x_1 \geq 0\,,\, x_2 \geq 0$
funkcija cilja	$Z\colon\ 6000\,x_1 + 9000\,x_2\ \rightarrow max$

Standardna forma je omogućava da se nejednačine pretvore u jednačine i pri njenom formiranju je neophodno poštovati pravilo:

ako je ograničenje $\leq$ sa leve strane se dodaje jedna slek promenljiva koja preuzima neiskorišćeni resurs pa je

Standardna forma:

površina	$x_1 + \quad x_2 + x_3 \qquad = 7$
pesticidi	$9\,x_1 + 3\,x_2 \quad + x_4 \qquad = 64$
đubrivo	$3\,x_1 + 6\,x_2 \qquad + x_5 \quad = 42$
uslov nenegativnosti	$x_1 \geq 0\,,\, x_2 \geq 0$
funkcija cilja	$Z\colon\ 6000\,x_1 + 9000\,x_2 + 0\,(x_3 + x_4 + x_5 + x_6) \rightarrow max$

prva simpleks tabela se formira upisivanjem brojeva iz standardne forme u tabelu pa je

Linija	Baza	Vrednost	x1	x2	x3	x4	x5
1	x3	7	1	1	1	0	0
2	x4	64	9	3	0	1	0
3	x5	42	3	6	0	0	1
4	/	0	6000	9000	0	0	0

Po pravilima, prvo se određuje radna kolona (ona u kojoj je u zadnjem redu najveći pozitivan broj) a to je kolona x2, a zatim i radni red (gde je količnik kolone Baza i radne kolone najmanji), odnosno 1. red proglašavamo radnim redom.

Linija	Baza	Vrednost	x1	x2	x3	x4	x5	
1	x3	7	1	1	1	0	0	/(-3) / (-6) /(-9000)
2	x4	64	9	3	0	1	0	
3	x5	42	3	6	0	0	1	
4	/	0	6000	9000	0	0	0	

Radni red se množi odgovarajućim koeficientima i rezultat je druga simpleks tabela:

Linija	Baza	Vrednost	x1	x2	x3	x4	x5
1	x2	7	1	1	1	0	0
2	x4	43	6	0	-3	1	0
3	x5	0	-3	0	-6	0	1
4	/	-63000	-3000	0	-9000	0	0

Pošto u zadnjem redu nema pozitivnih brojeva zaključuje se da je dobijen optimalni rezultat a to je x2=7 i maksimalni profit Z=63000din.

Kada se ovaj problem unese u program Lingo koji je specijalizovan za probleme optimizacije dobija se sledeći ekran

```
Global optimal solution found.
Objective value:                              63000.00
Infeasibilities:                              0.000000
Total solver iterations:                             3
Elapsed runtime seconds:                          0.03

Model Class:                                        LP

Total variables:                   2
Nonlinear variables:               0
Integer variables:                 0

Total constraints:                 4
Nonlinear constraints:             0

Total nonzeros:                    8
Nonlinear nonzeros:                0



  Variable            Value         Reduced Cost
```

```
    X1        0.000000           0.000000
    X2        7.000000           0.000000

   Row    Slack or Surplus      Dual Price
    1         63000.00           1.000000
    2         0.000000           3000.000
    3         43.00000           0.000000
    4         0.000000           1000.000
```

Zadatak 3: Data je tabela sa podacima

	Proizvod A	Proizvod B	Proizvod C	Kapacitet
Mašina 1	4	5	4	4200
Mašina 1	2	2	1	1500
Sirovina	2	3	4	2400
Dobit	10	12	10	

Optimizovati problem proizvodnje simpleks metodom tako da se ostvari najveća dobit.

Rešenje :

prvo se formiraju ograničenja

mašina 1 $4x_1 + 5x_2 + 4x_3 \leq 4200$

mašina 2 $2x_1 + 2x_2 \quad x_3 \leq 1500$

sirovina $2x_1 + 3x_2 + 4x_3 \leq 2400$

uslov nenegativnosti $x_1 \geq 0, x_2 \geq 0, x_3 \geq 0$

funkcija cilja $Z\colon\ 10x_1 + 12x_2 + 10\text{x}3 \rightarrow max$

Standardna forma je

mašina 1 $4x_1 + 5x_2 + 4x_3 + x_4 = 4200$

mašina 2 $2x_1 + 2x_2 \quad x_3 \quad + x_5 = 1500$

sirovina $2x_1 + 3x_2 + 4x_3 + \quad + x_6 = 2400$

uslov nenegativnosti $x_1 \geq 0, x_2 \geq 0, x_3 \geq 0$

funkcija cilja $Z\colon\ 10x_1 + 12x_2 + 10x_3 + 0(x_4 + x_5 + x_6) \rightarrow max$

Prva simpleks tabela je

Linija	Baza	Vrednost	x1	x2	x3	x4	x5	x6	
1	x4	4200	4	5	4	1	0	0	
2	x5	1500	2	2	1	0	1	0	/(-2.5) / (-1.5) /(-6)
3	x6	2400	2	3	4	0	1	1	
4	/	0	10	12	10	0	0	0	

Druga simpleks tabela je

Linija	Baza	Vrednost	x1	x2	x3	x4	x5	x6
1	x4	450	-1	0	3/2	1	-5/2	0
2	x2	750	1	1	1/2	0	1/2	0
3	x6	150	-1	0	5/2	0	-3/2	1
4	/	-9000	-2	0	4	0	-6	0

/(0.6) / (-0.2) /(-1.6)

Treća simpleks tabela je

Linija	Baza	Vrednost	x1	x2	x3	x4	x5	x6
1	x4	360	-2/5	0	0	1	-16/10	-3/5
2	x2	720	6/5	1	0	0	8/10	-1/5
3	x3	60	-2/5	0	1	0	-3/5	2/5
4	/	-9240	-2/5	0	0	0	-18/5	-8/5

Pošto u zadnjem redu u kolonama gde su promenljive nema pozitivnih brojeva zaključujemo da je postupak doveo do optimalnog rešenja a to je x1=0, x2=720, x3=60 a maksimum je 9240.

Kada se problem reši u programu Lingo dobija se sledeći listing

```
Global optimal solution found.
Objective value:                              9240.000
Infeasibilities:                              0.000000
Total solver iterations:                             2
Elapsed runtime seconds:                          0.03

Model Class:                                        LP

Total variables:                     3
Nonlinear variables:                 0
Integer variables:                   0

Total constraints:                   4
Nonlinear constraints:               0

Total nonzeros:                     12
Nonlinear nonzeros:                  0


Variable           Value        Reduced Cost
      X1        0.000000           0.4000000
      X2        720.0000            0.000000
      X3        60.00000            0.000000

     Row    Slack or Surplus      Dual Price
       1        9240.000            1.000000
       2        0.000000            3.600000
       3        360.0000            0.000000
       4        0.000000            1.600000
```

Zadatak 4: Dat je sledeći problem optimizacije

$$1\,x_1 + 1\,x_2 + 1\,x_3 \leq 1$$
$$1\,x_1 + 1\,x_2 + 2x_3 = 2$$
$$3\,x_1 + 2\,x_2 + 1\,x_3 \geq 4$$
$$x_1 \geq 0,\; x_2 \geq 0,\; x_3 \geq 0$$

funkcija cilja $Z\colon\ 2\,x_1 + 3\,x_2 + 4\,x3 \rightarrow max$

Rešenje:

Standardna forma je

$$1\,x_1 + 1\,x_2 + 1\,x_3 + x_4 = 1$$
$$1\,x_1 + 1\,x_2 + 2\,x_3 + x_6 = 2$$
$$3\,x_1 + 2\,x_2 + 1\,x_3 - x_5 + x_7 = 4$$

uslov nenegativnosti $x_1 \geq 0,\; x_2 \geq 0,\; x_3 \geq 0,\; x_4 \geq 0,\; x_5 \geq 0,\; x_6 \geq 0,\; x_7 \geq 0$

funkcija cilja $Z\colon\ 2\,x_1 + 3\,x_2 + 4\,x_3 + 0\,x_4 + 0\,x_5 - Mx_6 - Mx_7 \rightarrow max$

Pošto se u standardnoj formi pojavljuju viškovne i veštačke promenljive mora se koristiti dvostepeni Simpleks metod. Potrebni vektori su:

Vektor promenljivih	$X^T = [x_1, x_2, x_3, x_4, x_5, x_6, x_7]$
Vektor počenih baznih promenljivih	$X_0^T = [x_4, x_6, x_7]$
Vektor vrednosti početnih baznih promenljivih	$B^T = [1, 2, 4]$

Matrica koeficienata u jediničnim ograničenjima	$\begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 2 & 0 & 0 & 1 & 0 \\ 3 & 2 & 1 & 0 & -1 & 0 & 1 \end{bmatrix}$
Vektor koeficienata u ciljnoj funkciji	$C^T = [2, 3, 4, 0, 0, -M, -M]$
Vektor koeficienata u ciljnoj funkciji uz početne bazne promenljive	$C_0^T = [0, -M, -M]$

Potrebno je izračunati matrični račun

$$C^T - C_0^{\,T} * A = [2, 3, 4, 0, 0, -M, -M] - [0, -M, -M] \quad * \quad \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 2 & 0 & 0 & 1 & 0 \\ 3 & 2 & 1 & 0 & -1 & 0 & 1 \end{bmatrix}$$

$$C^T - C_0^{\,T} * A = [2 \;\; 3 \;\; 4 \;\; 0 \;\; 0 \;\; -M \;\; -M] - [-4M \;\; -3M \;\; -3M \;\; 0 \;\; M \;\; -M \;\; -M]$$

$$C^T - C_0^{\,T} * A = [2+4M \quad 3+3M \quad 4+3M \quad 0 \quad -M \quad 0 \quad 0]$$

$$C_0^T * B = [0, -M, -M] \quad * \quad \begin{bmatrix} 1 \\ 2 \\ 4 \end{bmatrix} = -6M$$

Na osnovu ovog računanja prva simpleks tabela je

Linija	Baza	Vrednost	x1	x2	x3	x4	x5	x6	x7
1	x4	1	1	1	1	1	0	0	0
2	x6	2	1	1	2	0	0	1	0
3	x7	4	3	2	1	0	-1	0	1
4	/	-6M	2+4M	3+3M	4+3M	0	-M	0	0

Zbog veće preglednosti zadnje red se deli u dva reda. Bira se najveći pozitivan broj u zadnjem redu i proglašava za pivot kolonu. Pivot red će biti linija broj 1.

Druga simpleks tabela je

Linija	Baza	Vrednost	x1	x2	x3	x4	x5	x6	x7
1	x4	1	1	1	1	1	0	0	0
2	x6	2	1	1	2	0	0	1	0
3	x7	4	3	2	1	0	-1	0	1
4	/	0	2	3	4	0	0	0	0
		-6	4	3	3	0	-1	0	0

/(-1) / (-3) /(-2)/(-4)

Po utvrđenom kriterijumu se formira treća simpleks tabela

Linija	Baza	Vrednost	x1	x2	x3	x4	x5	x6	x7
1	x1	1	1	1	1	1	0	0	0
2	x6	1	0	0	1	-1	0	1	0
3	x7	1	0	-1	-2	-3	-1	0	1
4	/	-2	0	1	2	-2	0	0	0
		-10	0	-1	-1	-4	0	-1	0

Pošto je linija broj 1 ispala iz baze rešenja a u njoj se nalazila promenljiva X_4, onda se briše kolona u kojoj se nalazi ta promenljiva pa nastaje

Četvrta simpleks tabela

Linija	Baza	Vrednost	x1	x2	x3	x5	x6	x7
1	x1	1	1	1	1	0	0	0
2	x6	1	0	0	1	0	1	0
3	x7	1	0	-1	-2	-1	0	1
4	/	-2	0	1	2	0	0	0
		-10	0	-1	-1	0	-1	0

Pošto u zadnjem redu u kolonama gde su promenljive nema pozitivnih brojeva zaključujemo da je postupak doveden do kraja. Predloženo rešenje je x1=1.

Potrebno je proveriti da li rešenje zadovoljava početna ograničenja

Uslov	Vrednost za $X_1=1$	Zadovoljen uslov
$1\ x_1 + 1\ x_2 + 1\ x_3 \leq 1$	1	Da
$1\ x_1 + 1\ x_2 + 2x_3 = 2$	1	Ne
$3\ x_1 + 2\ x_2 + 1\ x_3 \geq 4$	3	Ne

Kako rezultat simpleks metode ne zadovoljava početna ograničenja zaključuje se da problem nema rešenja i da za date početne uslove ne postoji realno rešenje.

4.3. Zadaci za vežbanje

Zadatak 1: Dat je linearni program:

Minimizirati: $z = x_1 + 2x_2$

pri ograničenjima: $x_1 + 3x_2 \geq 11$

$2x_1 + x_2 \geq 9$

uz uslov da su: x_1 i x_2 nenegativni.

a. Program prevesti u standardnu formu i zatim ga rešiti simpleks algoritmom.
b. Proveriti da li pri istim ograničenjima i uslovima gornji program ima maksimum.
c. Program rešiti i grafičkim metodom.

Zadatak 2: Rešiti metodom simpleks sledeći linearni program:

Minimizirati: $z = x_1 + x_2 + x_3 + x_4 + x_5 + x_6$

pri ograničenjima: $x_1 + x_6 \geq 7$

$x_1 + x_2 \geq 20$

$x_2 + x_3 \geq 14$

$x_3 + x_4 \geq 20$

$x_4 + x_5 \geq 10$

$x_5 + x_6 \geq 5$

uz uslov da su sve promenljive nenegativne.

Zadatak 3: Poljoprivredno gazdinstvo može da poseje pšenicu i kukuruz na najviše 19 ha. U magacinu se nalazi 46 l pesticida i 36 mc đubriva. Da bi se postigao solidan prinos pšenice treba uložiti 3mc đubriva i 8 l pesticida po svakom zasejanom hektaru. Za kukuruz ova ulaganja su 5 mc/ha đubriva i 3 l/ha pesticida. Ako je očekivani profit za pšenicu 20000 din/ha, a za kukuruz 24000 din/ha, odrediti na koliko je hektara optimalno posejati pšenicu, a na koliko kukuruz. Strukturu setve odrediti analitički, metodom Simpleks.

Zadatak 4: Poljoprivredna apoteka na raspolaganju ima 200 litara demineralizovane vode, 5 kilograma fosmeta i 4kg piriproksifena. Od raspoloživih hemijskih komponenti može da pravi tri proizvoda Imidan, Pyxal i Silico. Jedan kilogram Imidana donosi profit od 450 dinara, profit za jedan kilogram Pyxala je 600din, a profit po jednom kilogramu Silica je 550din. Za jedan kilogram zaštitnog sredstva pored tajnih sastojaka potrebno je za Imidan, 6 l demineralizovane vode, 100g fosmeta i 200g piriproksifena, za kilogram Pyxala 10 l demineralizovane vode, 200g fosmeta i 250g piriproksifena a za jedan kilogram Silica treba 8 l demineralizovane vode, 180g fosmeta i 400g piriproksifena.

Kako menadžer poljoprivredne apoteke treba da rasporedi raspoložive resurse (demineralizovanu vodu, fosmet i piriproksifen) na proizvode tako da bi apoteka ostvarila maksimalni profit?

Rešenje: Treba praviti samo Pyxal i to 16 kilograma jer se tako ostvaruje najveći profit od 9600din.

Zadatak 5: Neka je zadatak isti kao prethodni ali uz dodatni uslov da je, zbog asortimana i ponude potrebno da se pravi najmanje po 3 kilograma od svake vrste zaštitnog sredstva. Kakvo je onda optimalno rešenje?

5. LITERATURA

[1] Dantzig, G.B.: "Linear programming and extensions", 1998, Princeton University Press

[2] Gass, S.I.: "Linear Programming: Methods and Applications", Courier Dover Publications, 1985,

[3] J. Đorđević, Osnovi računarske tehnike, Zbirka rešenih zadataka , Blace 2005

[4] S. Krčevinac, M. Čangalović, V. Kovačević-Vujčić, M. Martić, M. Vujošević:Operaciona istraživanja 1, 3. izdanje, FON, Beograd 2009.

[5] Slobodan B. Vujić, Računarstvo i informatika, Mikro knjiga, 2001

[6] Srđević Bojan, Informatika, udžbenik, Poljoprivredni fakultet, Novi Sad, 1996.

[7] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein.Introduction to Algorithms, Second Edition. MIT Press and McGraw-Hill, 2001.

Dr Tihomir Zoranović

Primenjena informatika - zbirka zadataka