Strukture Podataka i Algoritmi

Lekcija 1: Uvod

leto 2019/2020

Prof. dr Branimir M. Trenkić

O meni....

- **Branimir M. Trenkić**
- Doktor tehničkih nauka, oblast – računarske i telekomunikacione mreže
- Redovni profesor Fakulteta za kompjuterske nauke “Megatrend” Univerziteta
- e-mail: trenkic.branimir@gmail.com
  - Molim da subject- linija Vašeg e-mail-a počinje sa kodom kursa (***SPA20***)

O kursu....

- Obim: **2** + **2**
- Termini:
  - ***Predavanja: ponedeljak 11:00 – 12:45, Sala K1***
  - Vežbe: naknadno će biti objavljen raspored vežbi
- Okosnica kursa:
  1. ***Podaci u različitim formama*** (ATP – ***apstraktni tipovi podataka***)
  2. ***Implementacija ATP-ova*** kroz ***strukture podataka***,
  3. ***Algoritmi*** osnovnih operacija definisanih nad ATP

O kursu....

- Način polaganja:
- ***Kroz predispitne obaveze***:
  - Odbrana vežbi (30 poena)
  - Aktivnost na nastavi (10 poena)
  - ***Teorija*** - Kolokvijum1 (30 poena) + Kolokvijum2 (30 poena)
  - U junskom roku je omogućeno kompletirati predispitne obaveze (**ali samo jedan deo!**)
- ***Završni ispit*** (u ispitnom roku)
  - Preduslov: Odbrana vežbi (30 poena)
  - Ispit se polaže pismeno (70 poena)
- Ispit je položen osvajanjem ***> 50 poena***

Osnovni materijal kursa

Udžbenik:

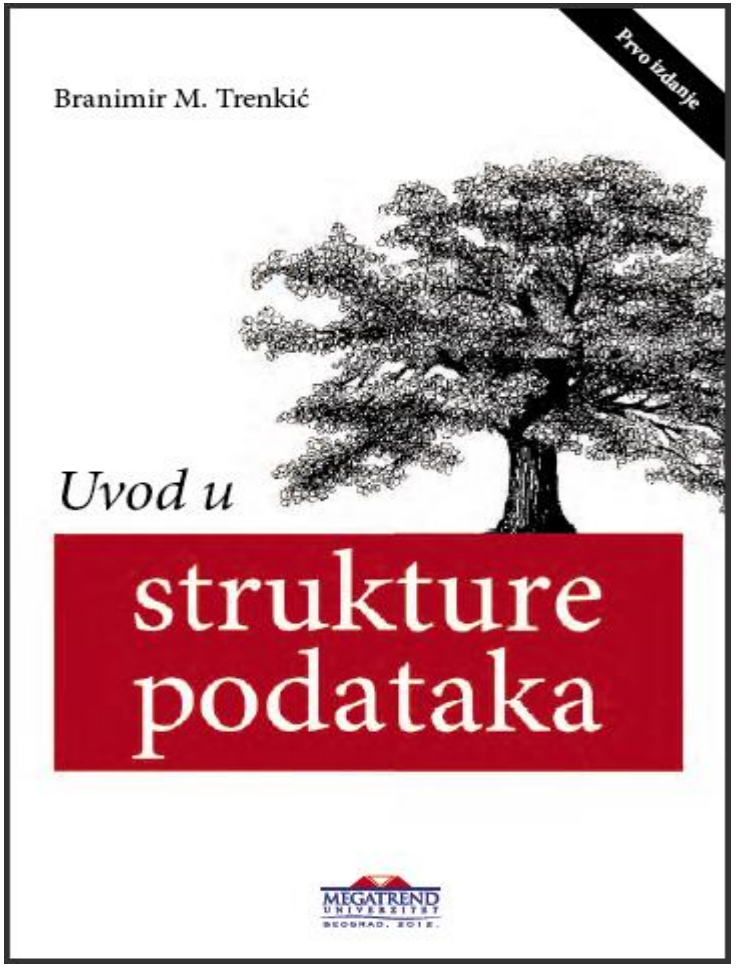
Branimir M. Trenkić

Uvod u strukture podataka

Prvo izdanje,

Megatrend univerzitet, 2013

ISBN 978-86-7747-485-0



Osnovni materijal kursa

Prezentacije sa predavanja **(***pdf format***)**

(Distribucija po dogovoru!)

Uvod

Niklaus Emil Wirth (1975.):

ALGORITHMS + DATA STRUCTURES = PROGRAMS

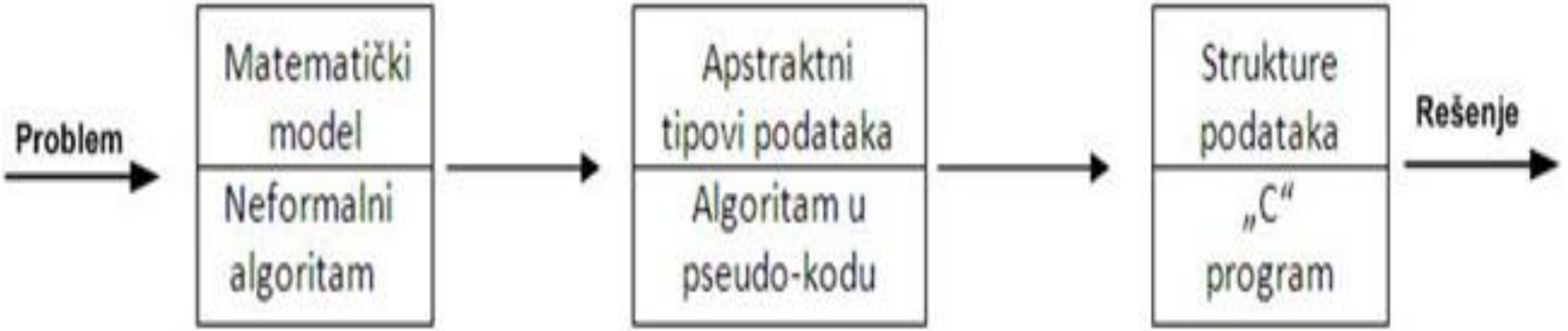
- Algoritami i strukture podataka – ***fundamentalni predmet izučavanja*** u računarstvu
- Računarski sistemi:
  - ***Smeštanje podataka*** u memoriji računara
  - ***Manipulacija podacima*** iz memorije
    - Skup instrukcija na osnovu kojih računar vrši obradu podataka - ***program***

Uvod

- U najopštijem smislu,
- ***Struktura podataka*** – termin koji opisuje ***način organizacije podataka u programu***
- ***Algoritam*** – ***postupak obrade podataka (definiše logiku programa)***
- ***Strukture podataka + Algoritmi*** – tesno povezani ***gradivni elementi*** od kojih se sastoji ***program***
- ***FKN*** – 2 predmeta
  1. Strukture podataka i algoritmi
  2. Dizajn i analiza algoritama

Računarski problem

- Problem za čije rešavanje je potrebno kreirati alat u vidu ***računarskog programa***
- Proces iz ***više faza*** – do konačnog alata



Podaci

- Na samom početku razjasniti ***pojam podataka***
- U kontekstu termina:
  1. ***Tip podataka***
  2. ***Apstraktni tip podataka*** (ATP ili ADT)
  3. ***Struktura podataka***

Tip podataka

- U terminologiji programskih jezika,
- ***Tip podataka - skup vrednosti*** koji neki entitet programa (promenljiva, konstanta, vraćena vrednost funkcije) - ***može imati***
- Na primer, ***promenljiva logičkog*** (*boolean*) ***tipa*** može imati samo dve vrednosti to su ***tačno*** (*true*) ili ***netačno*** (*false*)
- Za tip podataka su usko vezani i ***dozvoljeni operatori***

Tip podataka

- ***Osnovni*** (***elementarni***) ili ***primitivni*** tipovi podataka – objekti tog tipa ***zadovoljavaju svojstvo atomičnosti*** – ne mogu se razdvojiti na prostije celine
- Takvi tipovi podataka se javljaju kao standardni ili ***„ugrađeni“ tipovi*** podataka programskog jezika
- Oni variraju od programskoj jezika do programskog jezika; kao što smo već videli, ***u C jeziku*** to su *int*, *float* (*double*) i *char* tipovi

Apstraktni tip podataka

- Predstavljaju ***generalizaciju primitivnih tipova podataka*** i rezultat su ***primene principa apstrakcije*** u rešavanju računarskih problema
  - ***Apstrakcija problema*** – ***zanemarivanje nevažnih detalja*** koji ne utiču na rešenje
- Nezaobilazni princip u rešavanju složenijih problema (dekompozicija)

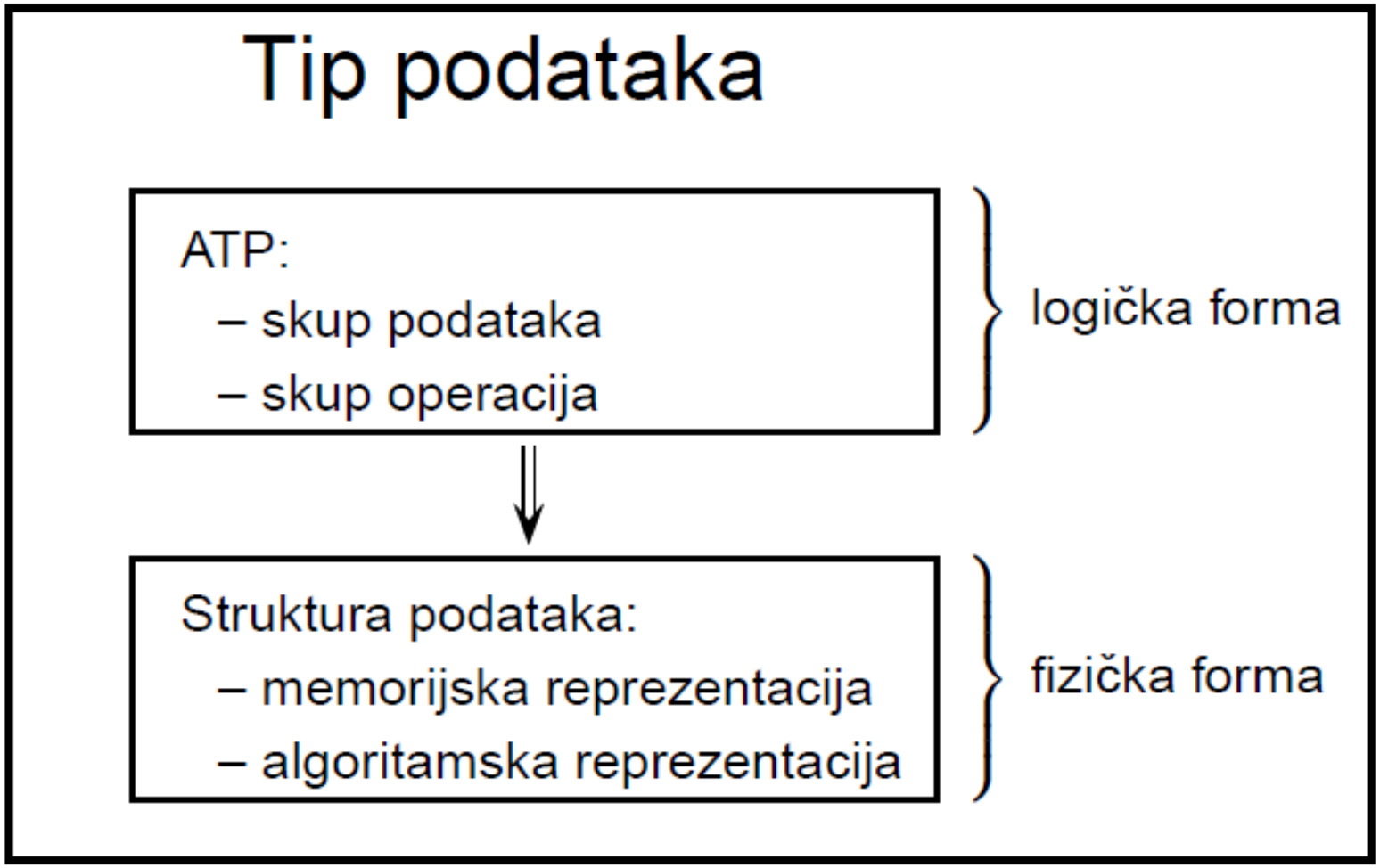
Apstraktni tip podataka

- Apstrakcija problema – ima ***dva aspekta***:
- ***Apstrakcija podataka***
  - ***Razdvajanje logičke slike podataka*** koji se obrađuju od njihove ***fizičke realizacije u računaru***
- ***Apstrakcija procedura***
  - ***Razdvajanje funkcionalnosti*** procedura od njihove ***programske realizacije*** u računaru

Strukture podataka

- ***Strukture podataka*** – sledeća forma podataka koja omogućuje ***implementaciju*** već specificiranog ***apstraktnog tipa podataka***
- Gradi od (***A***) ***osnovnih tipova podataka*** kao gradivnih elemenata korišćenjem (***B***) ***raspoloživih načina strukturiranja*** za njihovo međusobno povezivanje u logičku celinu
- Najčešće korišćeni ***agregacioni mehanizmi***:
  - Nizovi
  - Strukture ili zapis
  - Datoteke

Dva pojavna oblika podataka



Primer

- Program za ***simuliranje rada nekog restorana***
- Restoran na meniju ima ***jelo ražnjiće*** (***čisti tanjiri***)
- Jedan od modula programa simulira ***osnovne aktivnosti (funkcija) nad ovim objektom***
  1. Stavljanje komada mesa na žicu ražnjića
  2. Skidanje komada mesa sa žice prilikom konzumiranja
- ***Apstraktni tip podataka*** kojim se mogu opisati navedene operacije – ***STEK***
- U daljem postupku programske realizacije ***STEK kao struktura podataka (niz ili povezana lista?)***

Klasifikacija struktura podataka

- Osnovna podela struktura podataka - ***međusobnih relacija*** i ***veza*** elemenata u strukturi :
- ***Linearne*** strukture
  - ***U relaciji samo sa dva druga elementa strukture*** (***predhodnikom*** i ***sledbenikom***)
  - niz, povezana lista, stek, red čekanja
- ***Nelinearne*** strukture
  - Međusobni ***odnosi*** između elemenata ***složeniji*** (na primer, ***hijerarhijski***), u kojima jedan element može biti u vezi sa ***više drugih elemenata*** strukture
  - stabla i grafovi

Klasifikacija struktura podataka

- ***Kriterijum*** - mogućnost ***promene veličine*** strukture pri izvršavanju:
- ***Statičke*** i
  - Imaju ***fiksnu veličinu*** (dužinu) koja se određuje prilikom deklaracije strukture
- ***Dinamičke***
  - Mogu da se ***povećavaju i smanjuju*** u vreme ***izvršavanja programa*** saglasno aktuelnim potrebama za dodavanjem novih elemenata ili uklanjanje postojećih
  - ***Efikasnije*** - zauzimaju onoliko memorijskog prostora koliko je stvarno potrebno

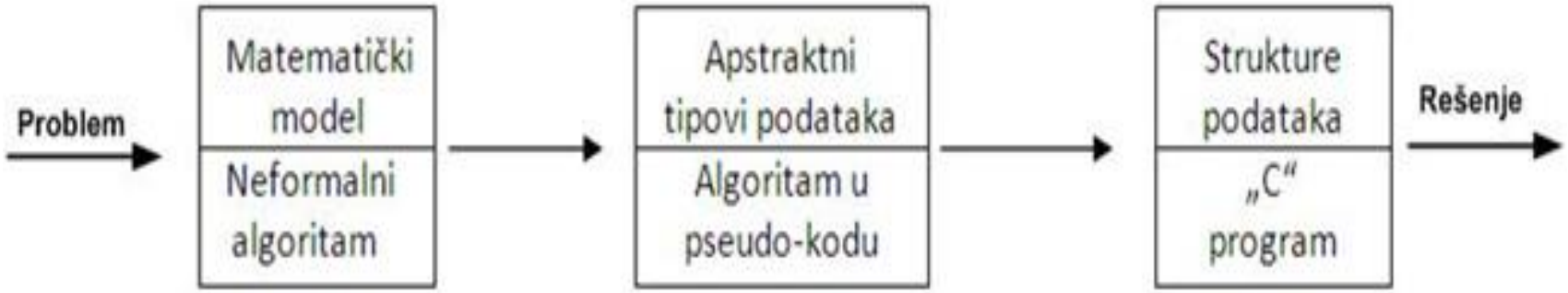
Memorijska reprezentacija

- Reprezentacija strukture podataka u memoriji se često naziva ***memorijskom strukturom***
- Na osnovu fizičkog i logičkog ***rasporeda elemenata*** strukture ***u memoriji***:

1. ***Sekvencijalna*** (***kontinualna***) reprezentacija i
2. ***Povezana*** (ulančana, spregnuta, ***ne-kontinualna***) reprezentacija

Algoritmi

- Koriste se za rešavanje ***računarskih problema*** i u tesnoj su vezi sa strukturama podataka u tom kontekstu
- Proces rešavanja računarskog problema:



- Šta se podrazumeva pod računarskim problemom?

Algoritmi

- Da bi se proces rešavanja računarskog problema mogao uspešno okončati - ***problem mora biti precizno definisan (zadat)***
- To je moguće uraditi na ***dva načina***:

1. Preciznom ***specifikacijom (opisom) ulazno-izlaznog odnosa***
2. U opštem slučaju koristi se ***formalniji (matematički) aparat*** – kako bi se otklonila svaka dvosmislenost

Algoritmi

1) Preciznom ***specifikacijom*** **(*opisom*) *ulazno-izlaznog odnosa*** kojim se ***izražava priroda problema***

- Problem ***sortiranja niza brojeva***
  - ***Ulaz*** – niz brojeva u proizvoljnom redosledu
  - ***Izlaz*** – niz istih brojeva poređanih u rastući redosled
- Problem ***usitnjavanja novčanog iznosa***
  - ***Ulaz*** – novčani iznos i apoeni metalnih novčića
  - ***Izlaz*** – minimalni broj tih novčića kojima se usitnjava dati novčani iznos

Algoritmi

2) Formalni način definisanja (zadavanja) problema

1.Ulazni parametri i

2.Odgovarajući ***izlazni parametri***

+ ***matematički model*** koji precizno definiše ulazne i izlazne parametare kao i njihov odnos

Algoritmi

2) Formalni način definisanja (zadavanja) problema

- U literaturi se često koristi pojam ***instanca problema*** koji podrazumeva skup konkretnih vrednosti za ulazne parametre, a pod ***rešenjem instance problema*** – podrazumevaju se odgovarajuće konkretne vrednosti koje čine rešenje, t.j. izlazne parametre
- Instanca problema sortiranja - ***[23,17,8,20,15]***
- Rešenje te instance - ***[8,15,17,20,23]***

Algoritmi

2) Formalni način definisanja (zadavanja) problema

Ulaz:

- pozitivan ceo broj ***k*** (broj metalnih novčića)
- niz od k celih brojeva ***[c_1,....c_k]*** (sitni apoeni)
- ***a*** novčani iznos

Izlaz:

- niz od k celih brojeva ***[b_1,....b_k]*** takav da važi:

$$a = \sum_{i=1}^{k} b_i \cdot c_i \qquad \sum_{i=1}^{k} b_i \quad \text{je minimalno}$$

Algoritmi

- Na zadati računarski problem – sledi definisanje ***matematičkog modela problema***
- Model se sastoji ***iz dva dela*** koji ***opisuju njegove ulazne i izlazne parametre***
- Oba dela se predstavljaju ***u vidu odgovarajućih matematičkih objekata*** kao što su brojevi, skupovi, funkcije itd...
- ***Ulazni deo*** - ***opšti model instance*** problema
- ***Izlazni deo – opšti model rešenja instance*** problema

Algoritmi

- ***Nakon*** kreiranja odgovarajućeg ***matematičkog modela*** za dati problem - treba ***naći rešenje*** na bazi definisanog modela
- Početni ili ***među-korak*** je ***opisati rešenje*** – ***opis rešenja nazivamo algoritam***
- ***Šta je algoritam?***
- ***DEF.***- Algoritam je ***diskretan i jednoznačan postupak*** predstavljen ***konačnim*** brojem instrukcija (pravila) koji ***dovodi do rešenja*** nekog konkretnog ***problema*** iz nekog skupa ulaznih podataka u ***konačno mnogo koraka***

Zapis algoritma

- Postoji ***više načina*** kako se algoritam može ***predstaviti***
- Željeni stepen ***formalizma*** u predstavljanju
  1. Korišćenje ***govornog jezika*** u prikazu algoritma
  2. Korišćenjem nekog ***programskog jezika*** sa njegovim strogim sintaksnim i semantičkim pravilima
- Kompromisna rešenja:

1. Algoritamske šeme kao grafički način i
2. ***Pseudo jezici (ili, strukturni prirodni jezici)***

Zapis algoritma - NZD

- ***Problem***: Najveći zajednički delilac (NZD)

Zapis algoritma - NZD

- ***Problem***: Najveći zajednički delilac (NZD)
- Pseudo kod :

```
//Ulaz: pozitivni celi brojevi x i y

//Izlaz: nzd(x,y)

algorithm nzd(x,y)

      d = min{x,y};

      while((x % d != 0)||(y % d != 0))do

          d = d - 1;

      return d;
```

Zapis algoritma - ***NZD***

Euklidovo rešenje: ***nzd(x, y) = nzd(y, x%y)***

Primer: ***nzd(12378, 3054) = ?***

- ***Rešenje:***

  12378, 3054

  12378, 3054, 162 (***12378 % 3054*** = 162)

  12378, 3054, 162, 138 (***3054 % 162*** = 138)

  12378, 3054, 162, 138, 24 (***162 % 138*** = 24)

  12378, 3054, 162, 138, 24, 18 (***138 % 24*** = 18)

  12378, 3054, 162, 138, 24, 18, 6 (***24 % 18*** = 6)

  12378, 3054, 162, 138, 24, 18, 6, 0 (***18 % 6*** = 0)

  ⇒ ***nzd(12378, 3054) = 6***

Zapis algoritma - NZD

- Pseudo kod (***Euklidov algoritam***, iterativno rešenje):

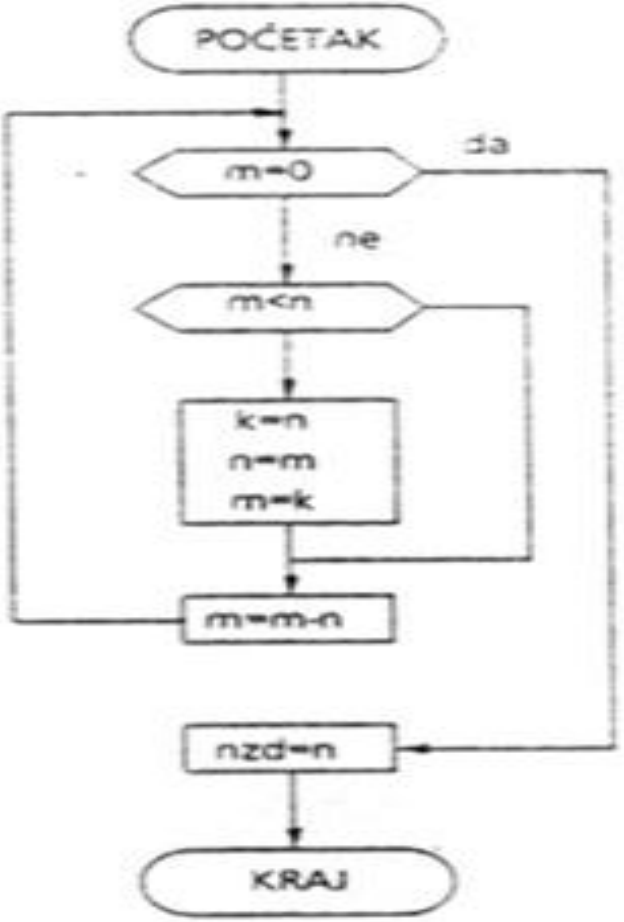
```
// Ulaz:  pozitivni celi brojevi x i y
// Izlaz: nzd(x,y)
algorithm euklid(x, y)
    while (y > 0) do
        z = x % y;
        x = y;
        y = z;
    return x;
```

Zapis algoritma - NZD

- Algoritamske šeme (dijagram toka):

Verzija sa oduzimanjem:

nzd(n,m) = ?



Algoritmi

- Predpostavke za ***uspešno rešenje*** datog problema:

1. ***Kreirati algoritam*** po izloženim principima
2. Dati problem mora biti ***algoritamski rešiv***

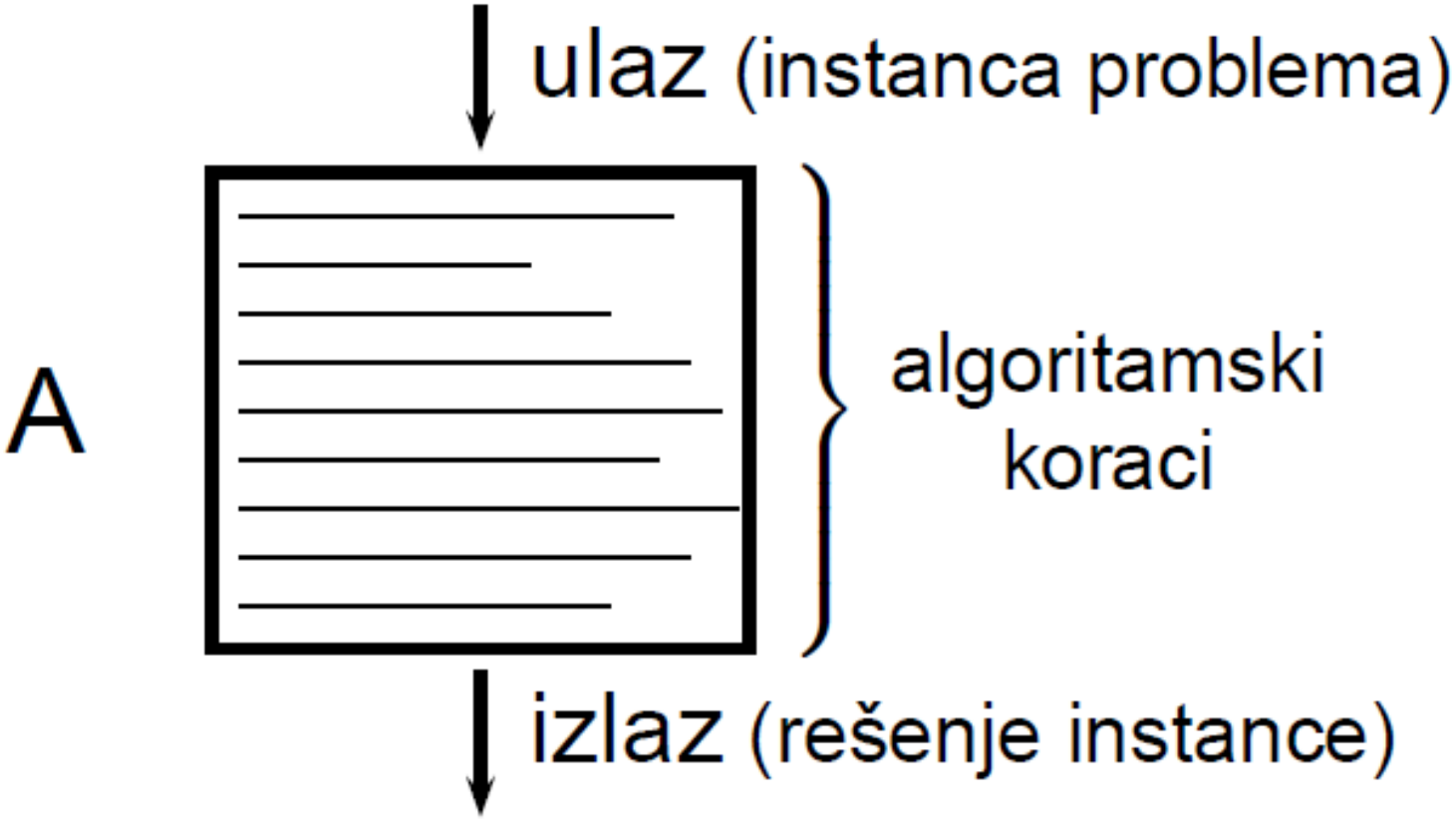
- Problem koji je algoritamski rešiv ne podrazumeva jedinstveno rešenje – takvi problemi ***mogu imati više rešenja*** (algoritama)
- ***Upoređivanje*** po kvalitetu algoritama koji rešavaju isti problem

Kriterijumi kvaliteta algoritma

- Korektnost (ispravnost)
- Vremenska složenost
- Prostorna složenost
- Jasnoća i jednostavnost
- Optimalnost

Algoritmi

- Analogija sa ***kuhinjskim receptom***
- Algoritam ≠ program



Algoritmi

- Precizan niz koraka koji dovode do rešenja datog problema
- Svi koraci se mogu mehanički izvršiti na računaru
- Analogija sa kuhinjskim receptom
- ***Algoritam ≠ program***