

ALGORITMI



ALGORITAM = UPUTSTVO = RECEPT

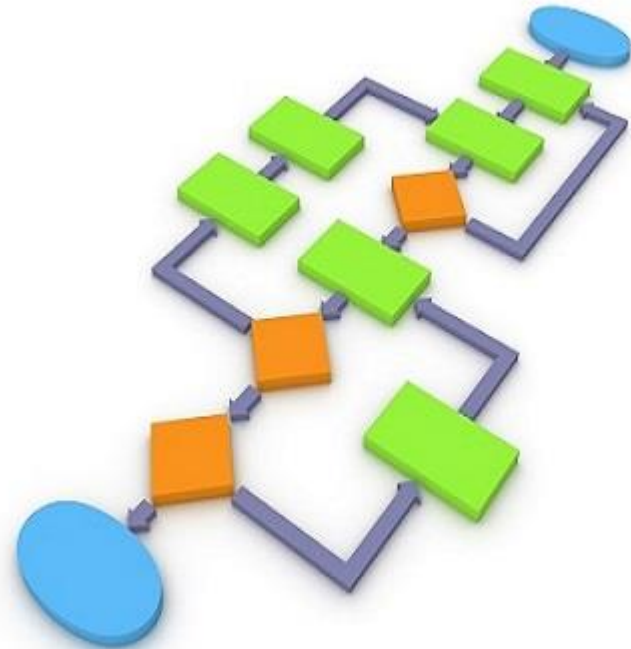


- **Algoritam** je precizno definisan postupak sa konačnom listom koraka za rešavanje nekog problema. **Algoritam** prihvata **ulazne vrednosti** i proizvodi **izlazne vrednosti**.

PRIMERI ALGORITAMA



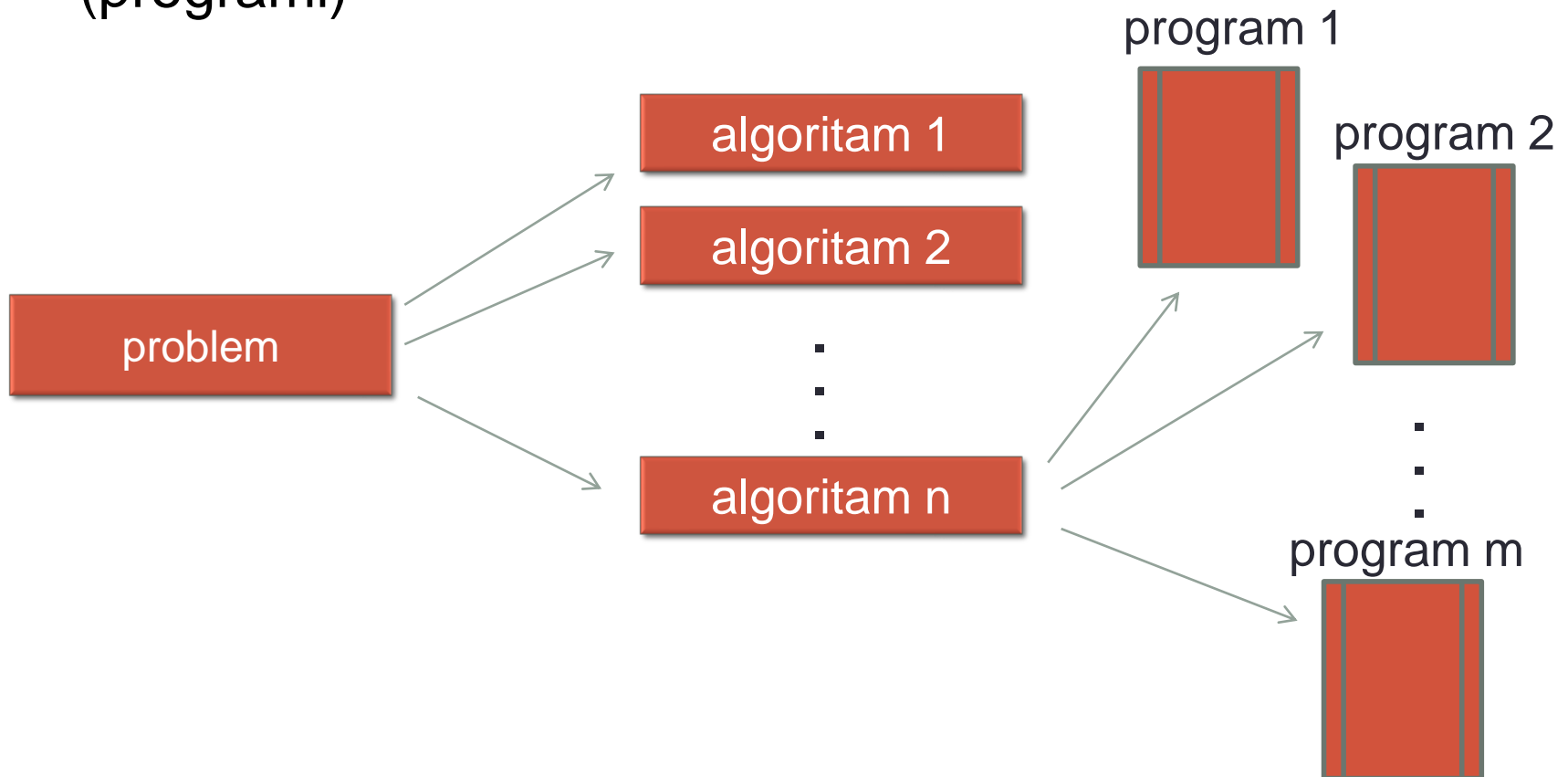
1. KUVANJE
2. KRETANJE PO GRADU
3. MNOŽENJE
4. ODREĐIVANJE NZD
5. PROGRAMIRANJE



PROBLEM, ALGORITAM I PROGRAM



- Različiti algoritmi kao rešenja istog problema
- Jedan algoritam može imati više različite implementacije (programi)



OSOBINE ALGORITMA



- Podrazumeva se da algoritam mora prvo da bude ispravan (tačan)!
 1. Diskretnost
 2. Determinisanost
 3. Efektivnost (konačnost)
 4. Rezultativnost
 5. Generičnost (masovnost)
 6. (Optimalnost)

DISKRETNOST



- Algoritam se sastoji od konačnog broja koraka
- Svaki korak zahteva obavljanje jedne ili više operacija
- U zavisnosti od operacija koje računar može da obavi, uvode se ograničenja za tip operacija koje se mogu koristiti u algoritmu

DETERMINISANOST



- Svaki algoritamski korak mora biti precizno definisan i potpuno jasan
 - Izrazi „13/0“ ili „oduzmi 4 ili 5 od trenutne vrednosti proizvoda“ nisu dozvoljeni
- Posle izvršavanja tekućeg koraka u algoritmu mora biti jednoznačno određeno koji je sledeći korak

REZULTATIVNOST



- Svaki algoritam mora posle konačnog broja koraka generisati traženi rezultat
- Algoritam može imati nula ili više ulaznih podataka, a može generisati jednu ili više izlaznih vrednosti

GENERIČNOST (MASOVNOST)



- Svaki algoritam definiše postupak za rešavanje klase problema, a ne pojedinačnog slučaja
- Pretraživanje ili sortiranje bilo kog podskupa celih ili realnih brojeva, množenje matrica/vektora bilo kog reda...

(OPTIMALNOST)



- Razlika između odličnog, dobrog, upotrebljivog i lošeg koda
- Ono što pravi razliku među programerima
- U nekim slučajevima nije prioritet
- U nekim slučajevima nije ostvarivo
- U nekim slučajevima izuzetno važno!

NAČINI PREDSTAVLJANJA ALGORITAMA



1. **Tekstualni opis na prirodnom jeziku**
2. **Grafički (pomoću dijagrama toka)**
3. Pseudokod
4. (Strukturogram)
5. Programski jezik

TEKSTUALNI OPIS ALGORITAMA



- Koriste se precizne rečenice govornog jezika
- Koristi se za lica koja se prvi put sreću sa pojmom algoritma
- Dobra osobina: razumljivost za širi krug ljudi
- Loše: nepreciznost koja proističe iz same prirode jezika



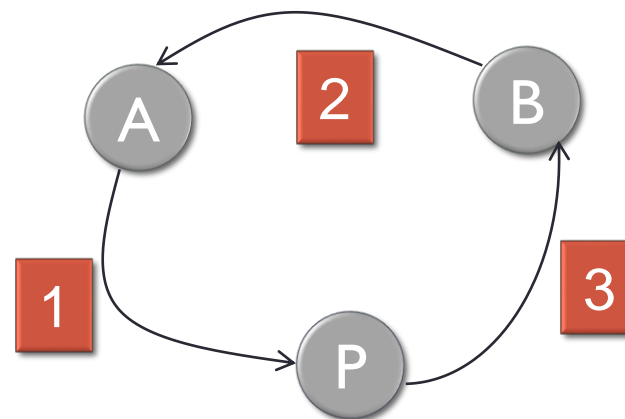
Euklidov algoritam za nalaženje **NZD** dva prirodna broja ***a*** i ***b***

1. Podeliti ***a*** sa ***b*** i ostatak zapamtiti u ***r***
2. Ako je ***r*** jednako **0**, **NZD** je ***b*** i došli smo do kraja algoritma, **inače** preći na **korak 3**
3. Zameniti ***a*** sa ***b***, ***b*** sa ***r***, i preći na **korak 1**



Zameniti sadržaje dve memorijske lokacije **A** i **B**

- Rešenje – potrebna je treća, pomoćna, lokacija
1. sadržaj lokacije **A** zapamtiti u pomoćnoj lokaciji **P**
 2. sadržaj lokacije **B** zapamtiti u lokaciji **A**
 3. sadržaj lokacije **P** zapamtiti u lokaciji **B**
 4. kraj





Pomeranje sadržaja lokacija ulevo

- Ciklički pomeriti u levo sadržaje lokacija **A**, **B** i **C**

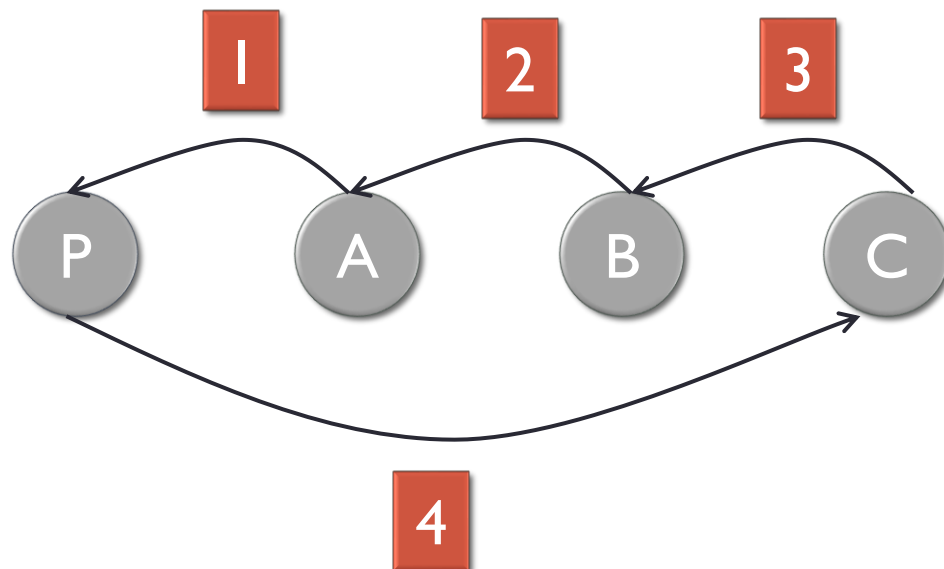
1. iz **A** u **P**

2. iz **B** u **A**

3. iz **C** u **B**

4. iz **P** u **C**

5. kraj

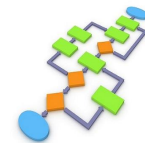


TEKSTUALNI OPIS ALGORITAMA – ZADATAK



1. Računanje maksimuma tri broja

GRAFIČKA PREDSTAVA ALGORITMA ...

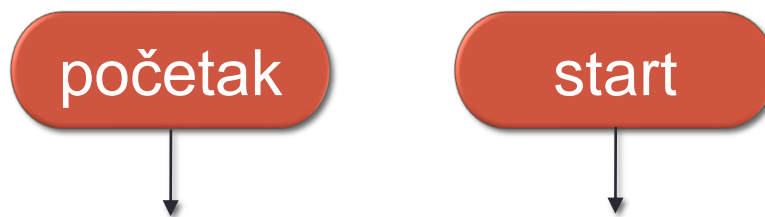


- Koriste se grafički simboli za predstavljanje pojedinih aktivnosti u algoritmu
- Ideja je pozajmljena iz teorije grafova
- Algoritam se predstavlja usmerenim grafom
 - čvorovi grafa predstavljaju aktivnosti koje se obavljaju u algoritmu
 - potezi ukazuju na sledeću aktivnost koja treba da se obavi

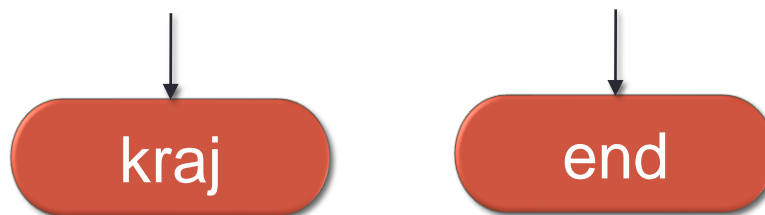
... GRAFIČKA PREDSTAVA ALGORITMA ...



- Polazni čvor u usmerenom grafu koji predstavlja algoritam nema dolaznih potega, a ima samo jedan izlazni poteg (granu)



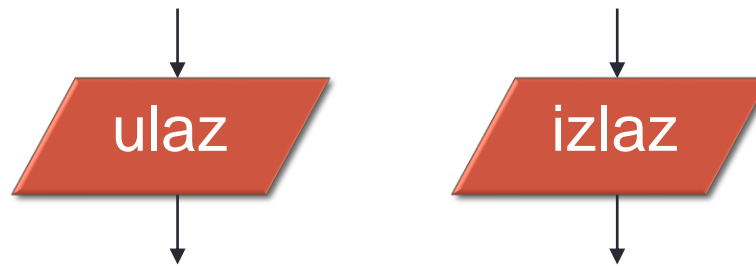
- Krajnji čvor u grafu koji predstavlja algoritam nema izlaznih potega, a ima samo jedan dolazni poteg



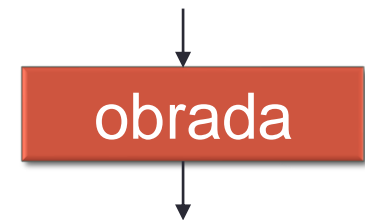
... GRAFIČKA PREDSTAVA ALGORITMA ...



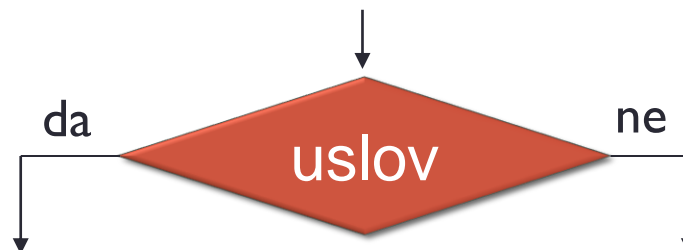
- Blok oblika romboida koristi se za označavanje ulaznih i izlaznih aktivnosti



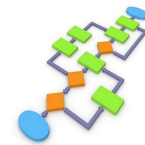
- Blok obrade je pravougaonog oblika



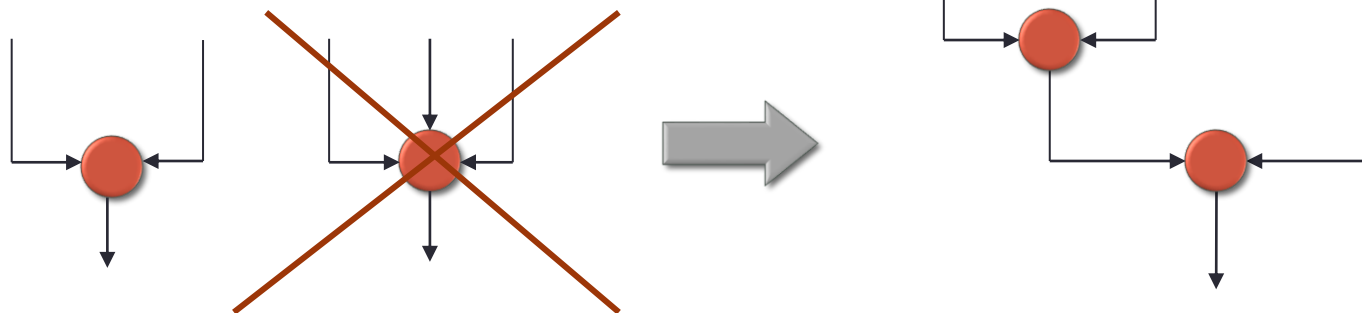
- Blok odluke



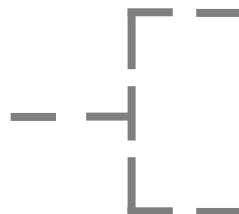
... GRAFIČKA PREDSTAVA ALGORITMA



- Blok spajanja potega



- Komentar



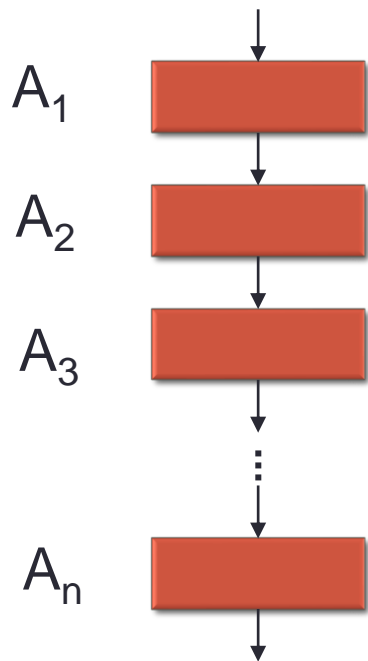
OSNOVNE ALGORITAMSKE STRUKTURE



- Kombinovanjem gradivnih blokova dobijaju se osnovne algoritamske strukture
 - sekvenca
 - selekcija (grananje, alternacija)
 - iteracija (petlja, ciklus)
- Pomoću osnovnih algoritamskih struktura može se konstruisati dijagram toka algoritma koji realizuje svaku izračunljivu funkciju (Teorema Bohm-Jacoppini 1966.)



- Linijska struktura koja se dobija kaskadnim povezivanjem blokova obrade

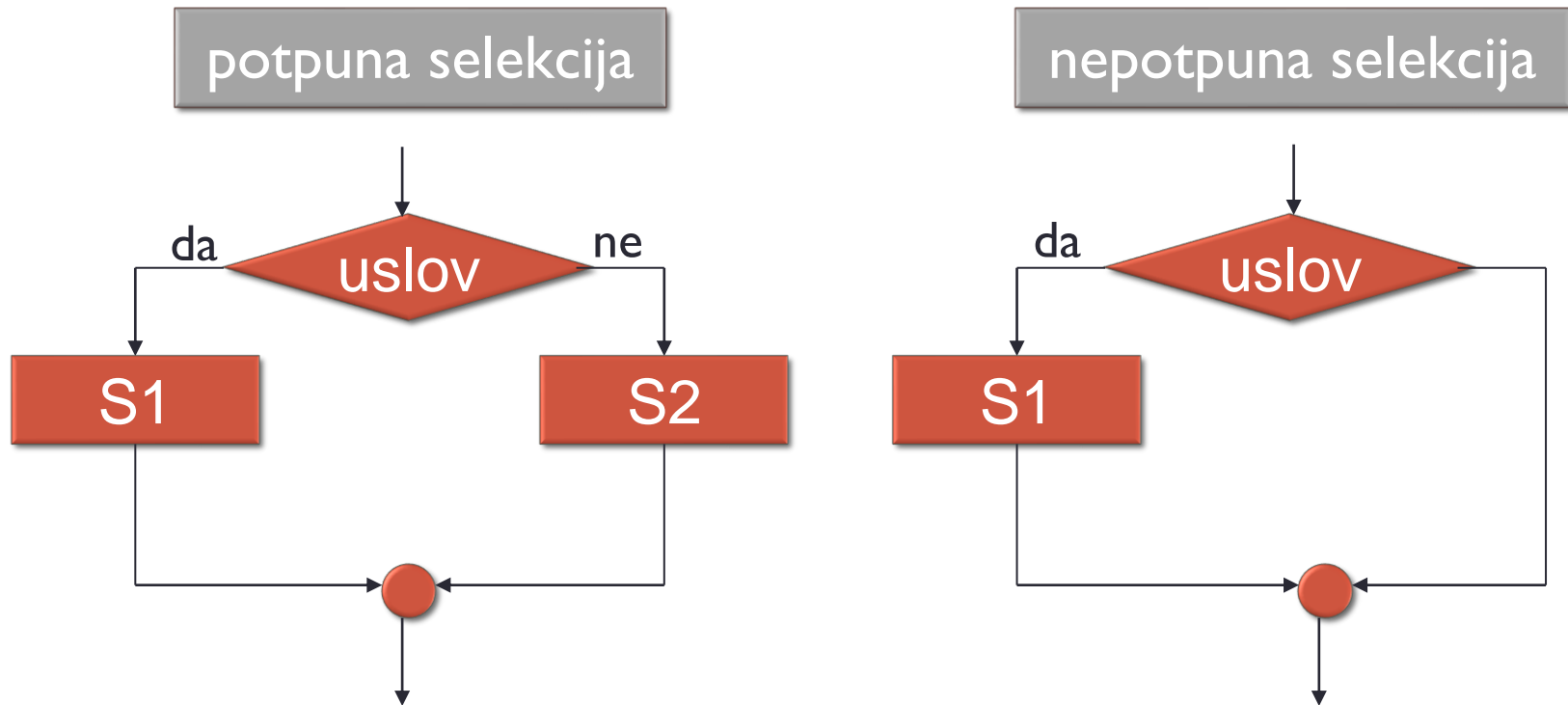


- Algoritamski koraci se izvršavaju redom, jedan za drugim
- Algoritamski korak A_i , $i = 2, \dots, n$ ne može da otpočne sa izvršenjem dok se korak A_{i-1} ne završi
- Sekvenca predstavlja niz naredbi dodeljivanja ($:=$)
- Oblik naredbe:
promenljiva:=vrednost
 $a:=b$
 $n:=n+1$

SELEKCIJA



- Linijska struktura koja se dobija kaskadnim povezivanjem blokova obrade

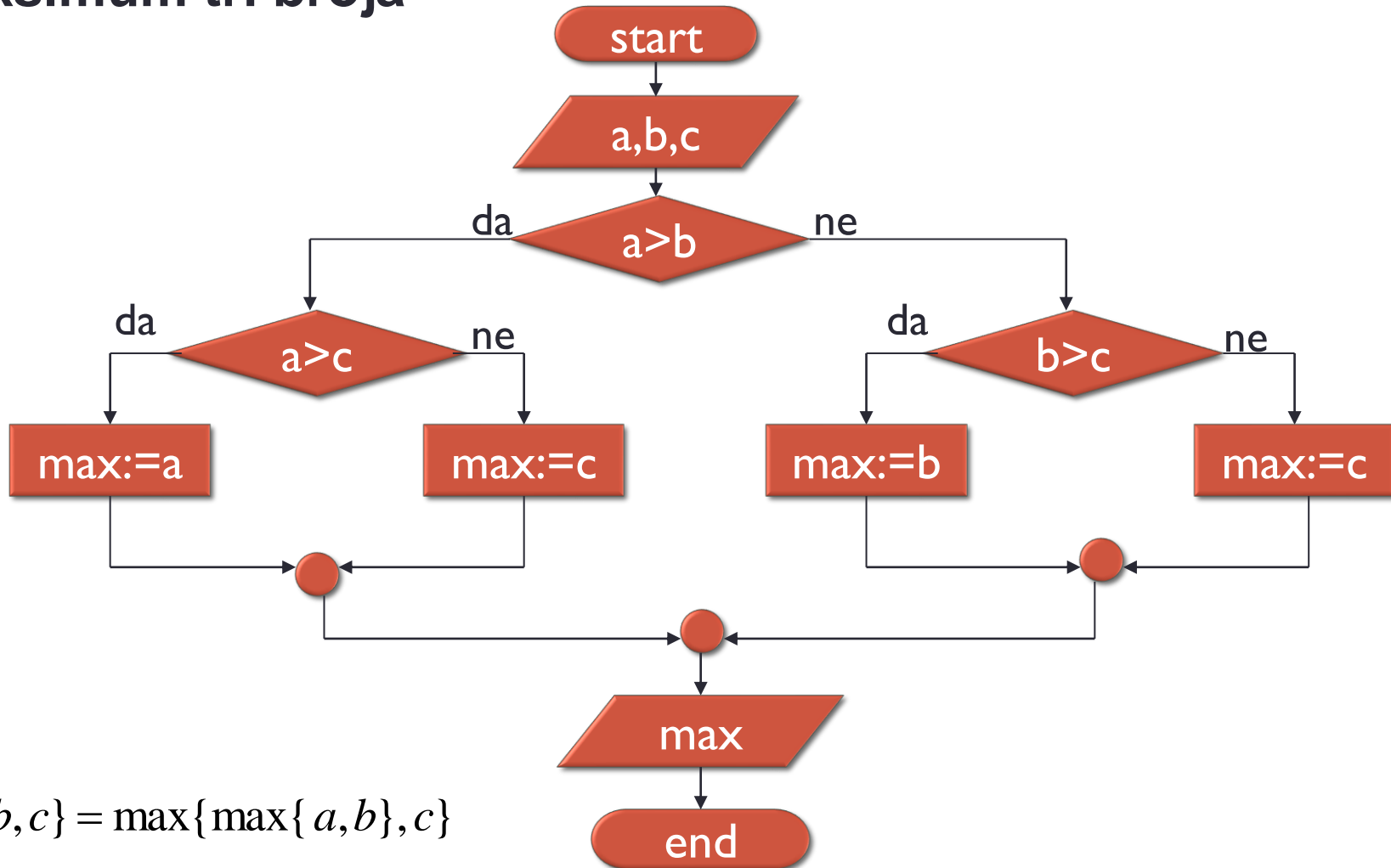


Blokovi označeni sa S1 i S2 mogu sadržati bilo koju kombinaciju osnovnih algoritamskih struktura.

SELEKCIJA – PRIMER



• Maksimum tri broja



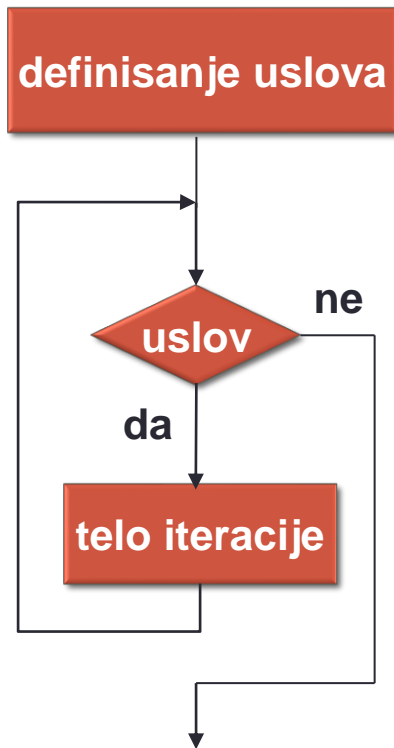
$$\max\{a, b, c\} = \max\{\max\{a, b\}, c\}$$

ITERACIJA

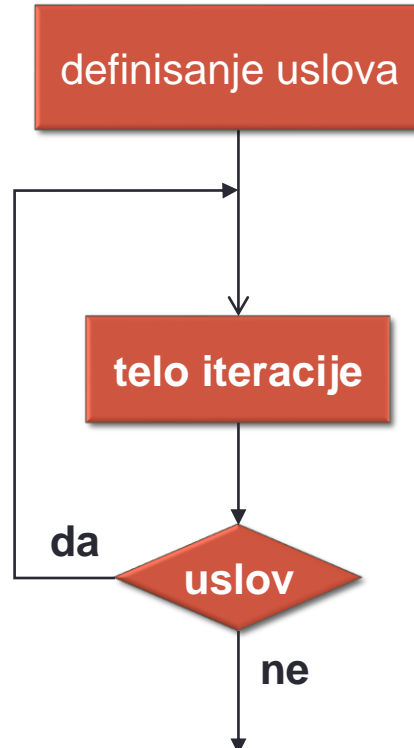


- Omogućava da se algoritamski koraci ponavljaju više puta

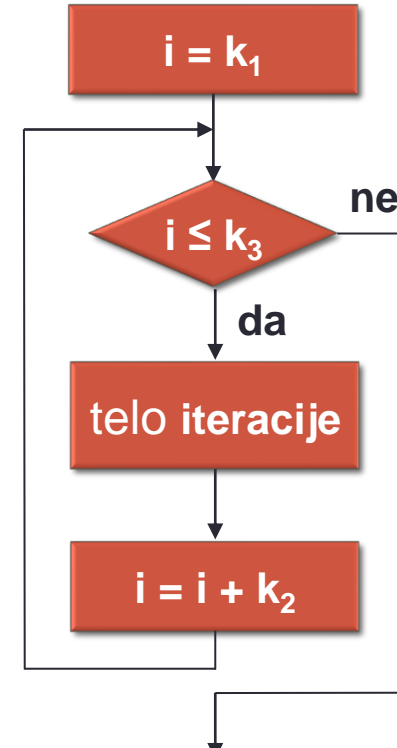
while



do-while



for (brojačka petlja)



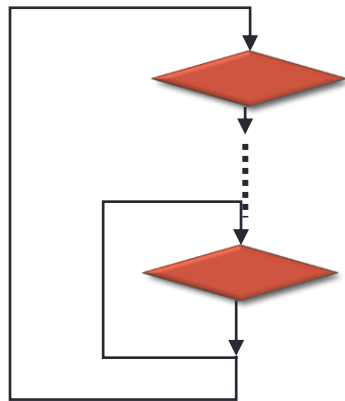
broj prolaza:

$$n = \left\lceil \frac{k_2 - k_1}{k_3} \right\rceil + 1$$

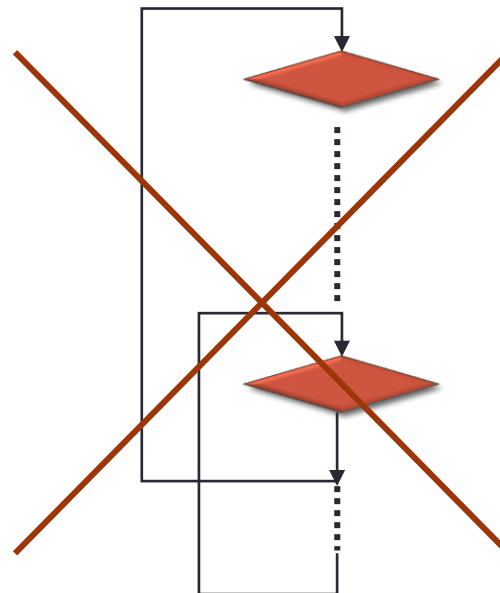
PRAVILA ZA ITERACIJA



- Ako iteracija počne unutar **then** bloka ili **else** bloka, u tom bloku se mora i završiti!
- Dozvoljene su paralelne (ugnježdene) iteracije.
- Nisu dozvoljene iteracije koje se seku!

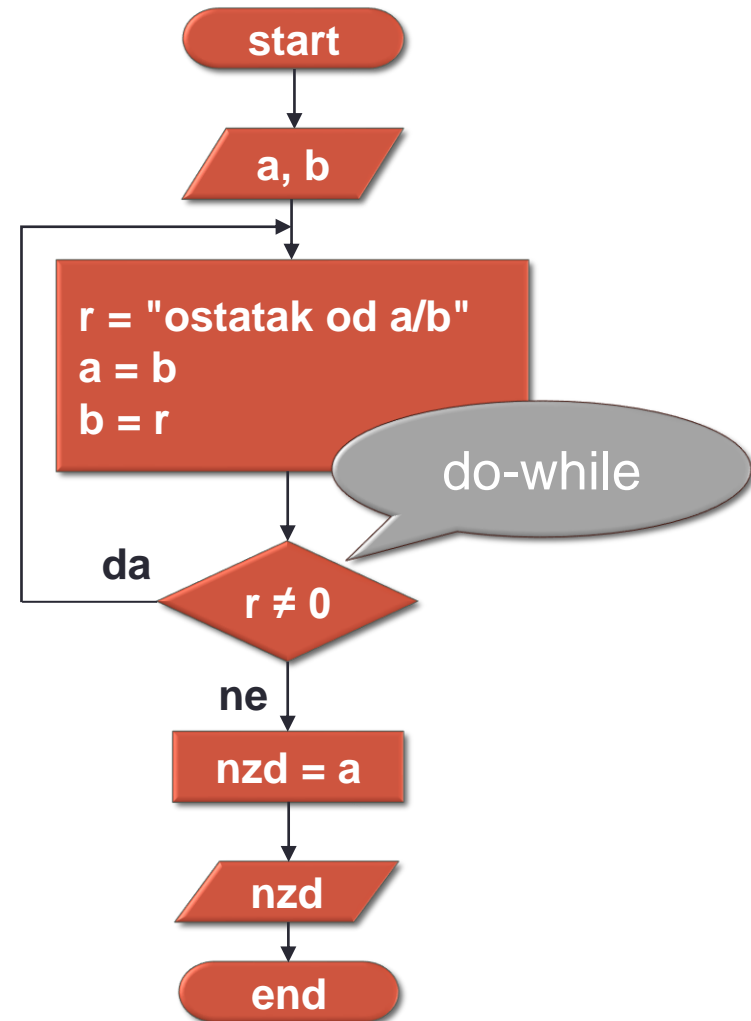
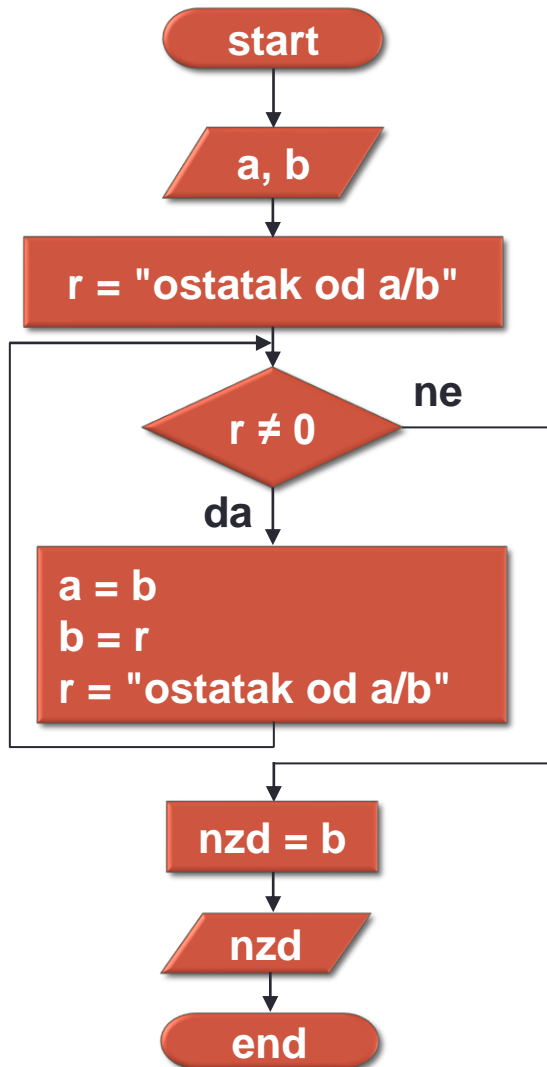


paralelne iteracije

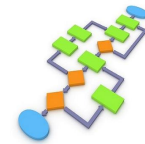


iteracije koje se seku

ITERACIJA – PRIMER



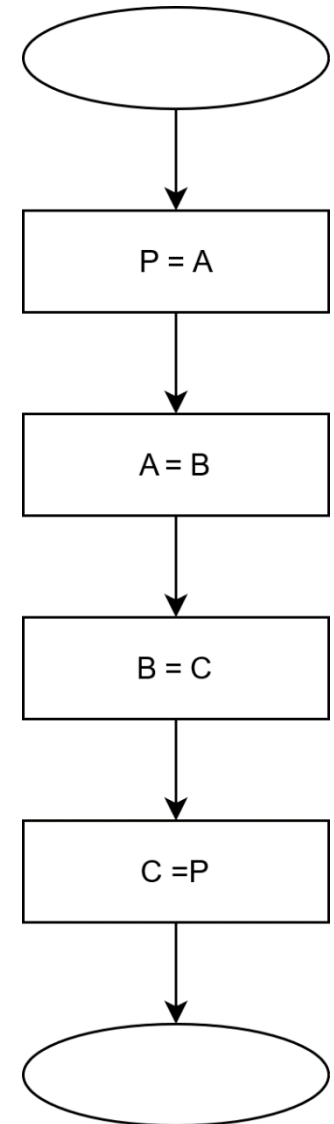
GRAFIČKA PREDSTAVA ALGORITAMA – ZADACI



1. Zameniti sadržaje dve memorijske lokacije A i B
2. Pomeranje sadržaja lokacija ulevo
3. Sabiranje prvih n prirodnih brojeva
4. Pronalaženje minimuma
5. Računanje sume tri broja
6. Računanje sume n brojeva



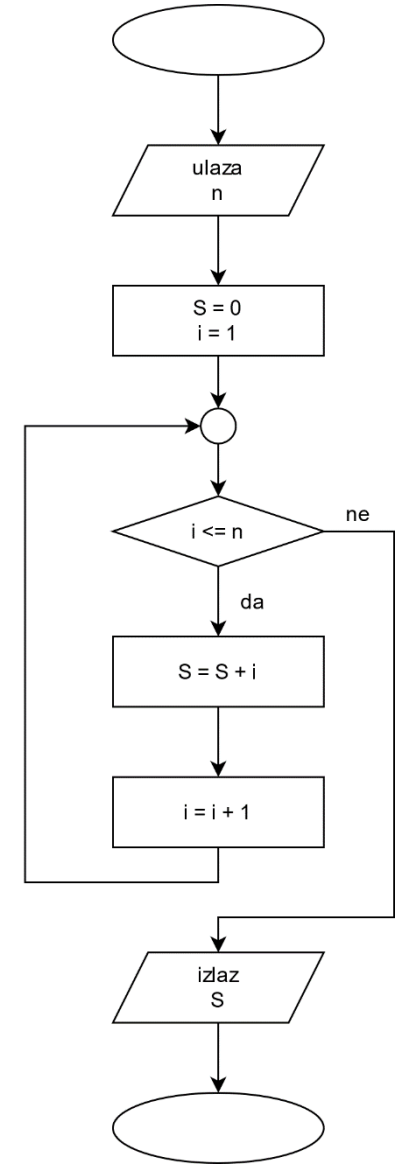
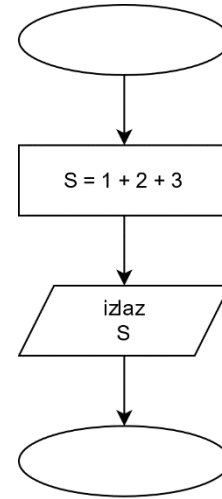
1. Zameniti sadržaje tri memorijske lokacije A i B



GRAFIČKA PREDSTAVA ALGORITAMA – ZADACI



3. Sabiranje prvih n prirodnih brojeva



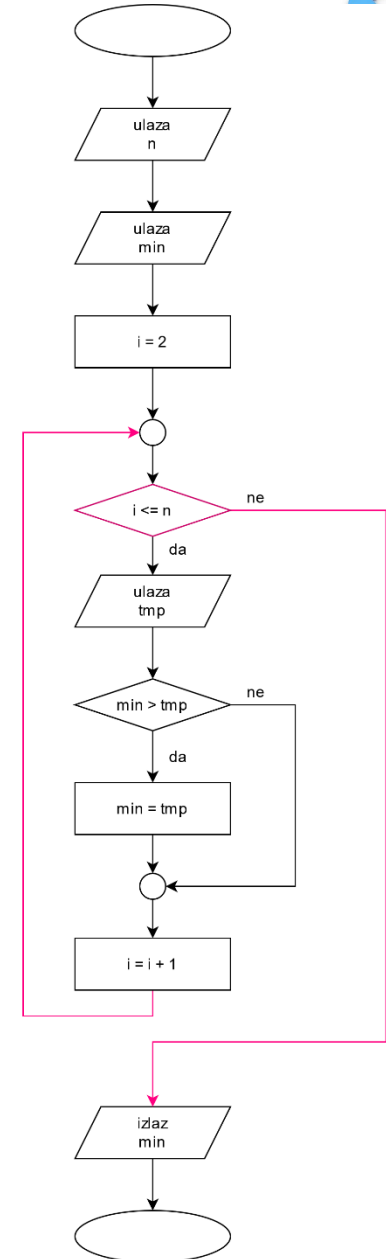
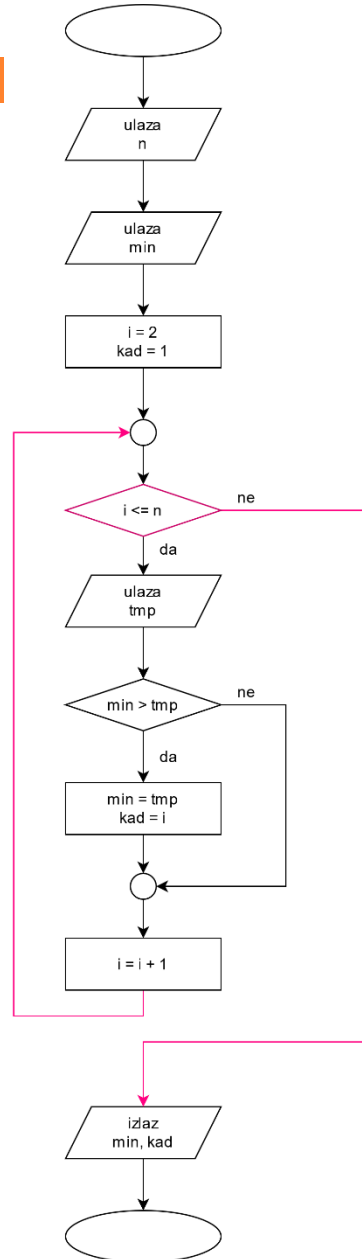
GRAFIČKA PREDSTAVA ALGORITAMA – ZADACI



Dragan de Dinu – Programiranje i programski jezici

U.VOĐ

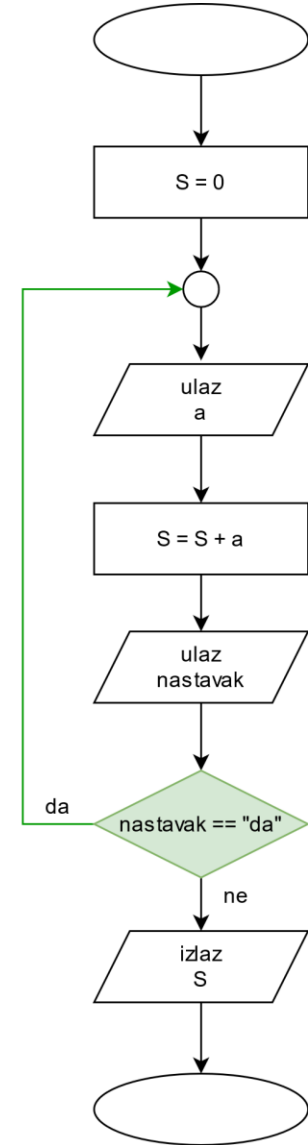
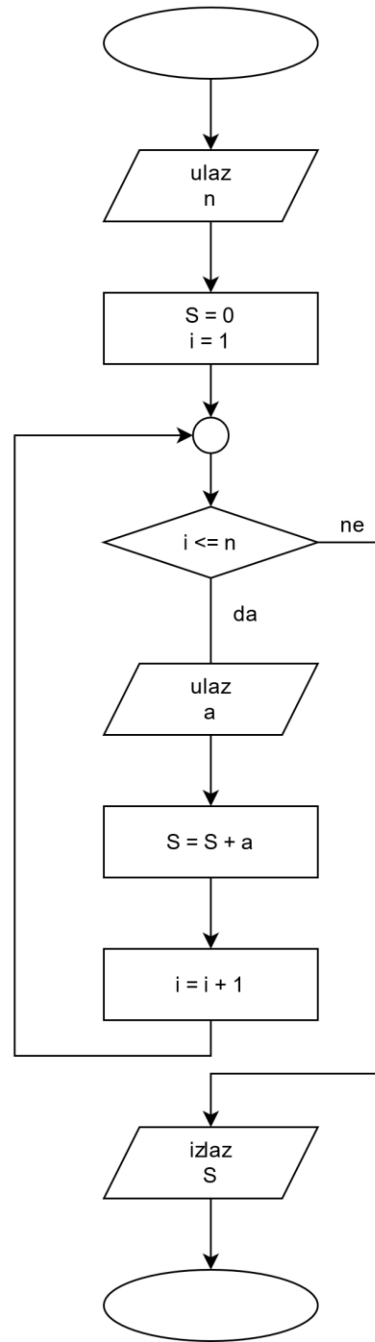
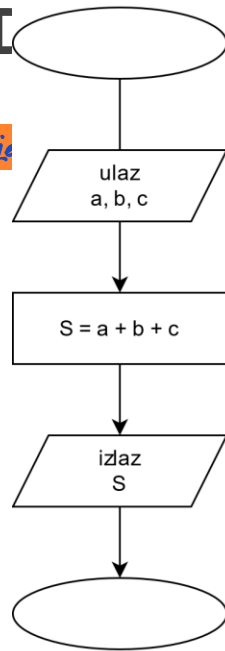
4. Pronalaženje minimuma



GRAFIČKA PRECI

Dragan de Dinu - Programiranje

- 5. Računanje sume tri broja
- 6. Računanje sume n brojeva



U.VOĐ

SLOŽENOST ALGORITAMA

SLOŽENOST ALGORITAMA



- Prilikom **projektovanja algoritama**, fokus je na **ispravnosti**
- **Analiza algoritama** ima za cilj da pruži **kvantitativnu ocenu njihove efikasnosti**
- Potrebno je analizirati kako radi algoritam korak po korak i na osnovu toga zaključiti koliko će biti **vreme njegovog izvršavanja u zavisnosti od broja ulaznih podataka** koje treba obraditi
- **Notacija veliko O** – izražava porast vremena izvršavanja algoritma u najgorem slučaju u zavisnosti od porasta veličine ulaznih podataka
- Pored **veliko O** notacije, postoji i notacije malo **o**, **omega** i **teta**

SLOŽENOST ALGORITAMA – PRIMERI ...



Notacija	Naziv vremena	Primeri algoritama
$O(1)$	konstantno	nalaženje većeg od dva broja
$O(\log N)$	logaritamsko	binarno pretraživanje niza
$O(N)$	linearno	linearno pretraživanje niza
$O(N \log N)$	linearno-logaritamsko	najbrži algoritmi za sortiranje, FFT
$O(N^2)$	kvadratno	jednostavni algoritmi za sortiranje
$O(N^3)$	kubno	množenje matrica
$O(C^N)$, $C > 1$	eksponencijalno (*)	rešavanje probl. putujućeg trgovca putem dinam. programiranja
$O(N!)$	faktorijelno (*)	rešavanje probl. putujućeg trgovca primenom metode grube sile

(*) za poslednje dve klase složenosti, ne možemo naći optimalno rešenje za veće instance problema – **kombinatorijalna eksplozija!** Ovi problemi nisu rešivi u polinomnom vremenu.

... SLOŽENOST ALGORITAMA – PRIMERI ...



- $O(N)$ – linearno pretraživanje niza (engl. linear search)
- Primer – tražimo broj 2 u sledećem nizu:

6 5 3 1 8 7 2 4
↑ ↑ ↑ ↑ ↑ ↑ ↑

- U najgorem slučaju trebaće 8 provera!
- Ako znamo da je niz uređen, možemo bolje!
- $O(\log N)$ – binarno pretraživanje niza (engl. binary search)

1 2 3 4 5 6 7 8
↑ ↑ ↑

- U najgorem slučaju trebaće 3 (tj. $\log_2 8$) provere!

... SLOŽENOST ALGORITAMA – PRIMERI



- $O(N^2)$ – sortiranje metodom zamenjivanja (metoda mehura – engl. bubble sort)
- Porede se susedni elementi, zavisno od izabranog uređenja (rastuće ili opadajuće), elementi, počev od najvećeg ka najmanjem ili obrnuto, “isplivavaju” na kraj niza, kao mehuri na površinu vode
- Algoritam jednostavan za implementaciju, ali uglavnom neefikasan
- Ostali algoritmi za sortiranje: insertion sort, selection sort, quick sort, heap sort, merge sort...
- Zanimljiv i koristan link: <http://bigocheatsheet.com/>

