

Pojam
i definicija
algoritma

Pojam algoritma

- Algoritam je precizno opisan postupak za rešavanje nekog problema
- Obično je to spisak uputstava ili skup pravila kojima je, korak po korak, opisan postupak za rešavanje zadatog problema
- Svaki korak algoritma, odnosno svako uputstvo iz spiska, mora da bude definisana operacija
- Algoritmi moraju da budu nedvosmisleni i da se završavaju u konačnom broju koraka

- Analiza, definicija i precizna identifikacija problema je veoma bitna faza programiranja, mada joj se veoma često ne posvećuje dovoljna pažnja
- Kada je to dovoljno precizno obavljeno i kada su jasno sagledani zahtevi prema računaru u rešavanju konkretnog problema pristupa se globalnom opisu algoritma na osnovu čega se kasnije detaljno izrađuje odgovarajući algoritam

- Sistematičnost i postupnost su veoma važni kod izrade algoritma kako bi kasnije pisanje programa bilo što jednostavnije
- Kada se program napiše i unese u računar sledi faza testiranja programa na računaru
- U toku ove faze uočavaju se i otklanjaju eventualne greške ili pojedina loša programska rešenja
- !! Ova faza može da bude i dugotrajna, pa čak i da nas vrati na prvu fazu ukoliko se pokaže da još u pripremnim radnjama nismo dovoljno dobro analizirali problem koji treba da rešimo

• Osnovne osobine algoritma:

– Konačnost

- Mora da dovede do rešenja primenom konačnog broja koraka. Ako broj koraka nije unapred poznat, treba predvideti postupak koji ograničava broj koraka

– Definisanost

- Svaki algoritamski korak mora da bude jednoznačno definisan. Treba predvideti sve slučajeve bez obzira na različite ulazne podatke

– Ulaz

- Algoritam može da ima jedan ili više ulaznih podataka, ali i ne mora da ih ima! Ulazni podaci se mogu uneti na početku, ali mogu i da se unose naknadno – u toku programa.

– Izlaz

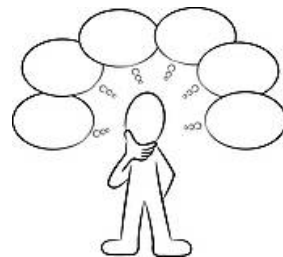
- Algoritam mora da ima jedan ili više izlaznih podataka

– Efikasnost

- Zahtev je da se dođe do traženog rešenja u što kraćem vremenu i uz primenu što manjeg broja algoritamskih koraka

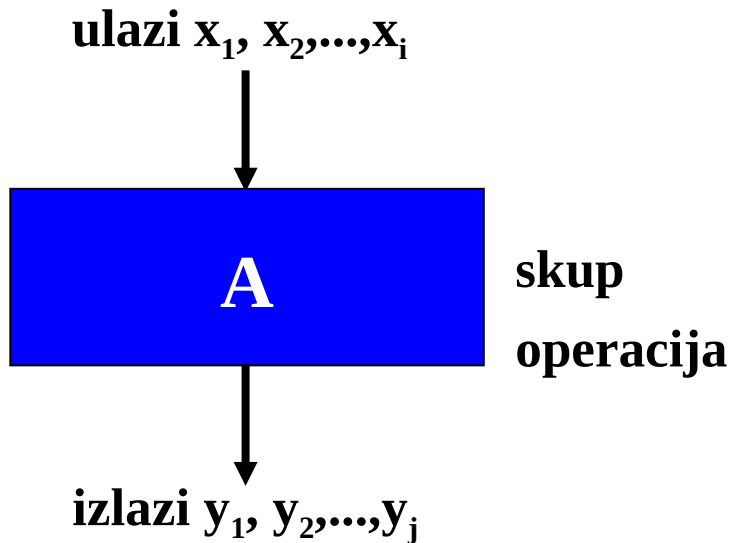
– Ostvarljivost

- Algoritam mora biti ostvarljiv na računaru!!!

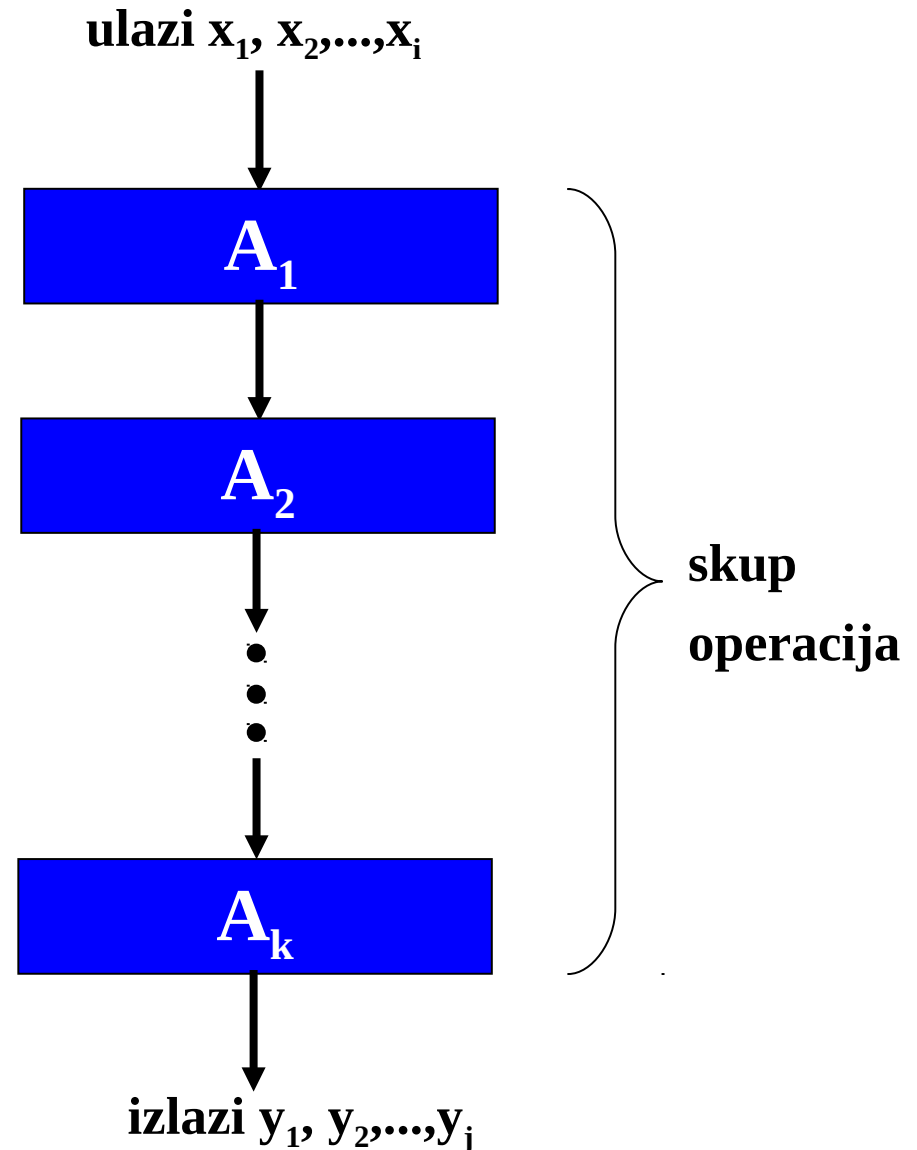


Algoritam

Opšta šema algoritma primenom dijagrama toka



Razlaganje algoritma na korake



Algoritam

- Algoritam je skup pravila formulisanih u cilju rešavanja određenog tipa zadatka
 - Algoritam se sastoji od niza **algoritamskih koraka**, čiji je redosled tačno utvrđen
 - Polazne veličine su **ulazne** veličine algoritma, a rešenja zadatka su **izlazne** veličine
- **Prikazivanje algoritma**
 - Prirodnim jezikom (razumljiv samo govornicima toga jezika)
 - Grafički (pomoću blok šema - Dijagramom toka)
 - Tekstualno (Primenom *pseudojezika*)
 - Odgovarajućim programskim jezikom

Algoritmi mogu biti prikazani ili realizovani na više načina:

- **Prirodnim jezikom** (razumljiv samo govornicima tog jezika)
- **Tekstualno - pseudokodom**, veštačkim precizno definisanim jezikom koji liči na programski jezik
- **Grafički**, dijagramom toka (blok-dijagramom) ili strukturnim dijagramima
- odgovarajućim **programskim jezikom**.

Prirodno-jezički opisi

Navedimo primer algoritma iz svakodnevnog života

– Recept za palačinke:

1. Uzmi dublju posudu!
2. Sipaj u nju 200g brašna i 2 jaja!
3. Ako imaš mleka, sipaj 2 dl mleka, u suprotnom sipaj 2dl vode.
4. Muti sve dok smesa ne bude bez grudvica.
5. Dodaj 1dl ulja.
6. Promešaj!

Prirodno-jezički opisi

Algoritam kuvanja čaja glasi:

- Uzeti lonče i sipati vodu.
- Uključiti ringlu.
- Sačekati dok voda ne proključa.
- Kad voda proključa, skinuti lonče i isključiti ringlu.
- Staviti kesicu čaja u lonče.
- Po želji, dodati kašiku šećera, mleko ili limun.
- Sipati čaj u šolju.

Opisi u obliku pseudo-koda

Pseudokod kombinuje običan, govorni jezik i preciznu, matematičku simboličku notaciju.

broj := unesi_broj

maksimum := broj

ponovi sledeće četiri puta

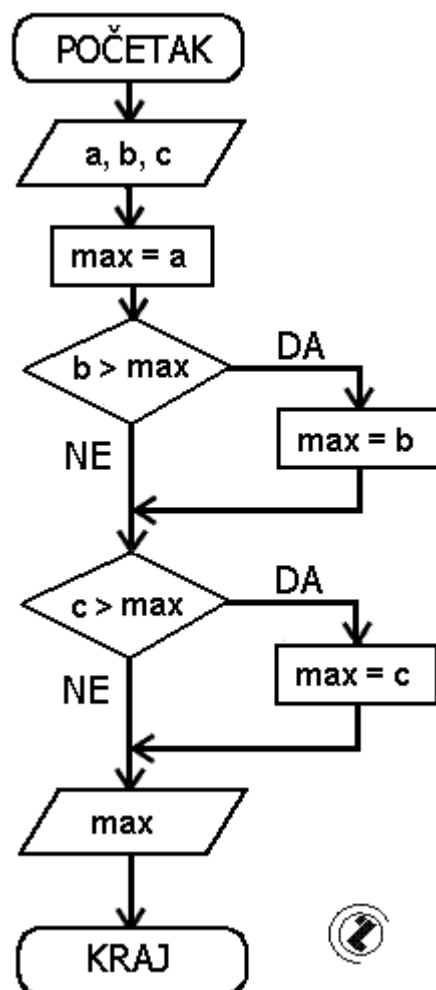
broj := unesi_broj

ako je broj > maksimum onda

maksimum := broj

saopšti maksimum

Grafički opis – dijagram toka



- Algoritam se opisuje dijagramom koji se sastoji od pojedinačnih blokova (predstavljenih pravougaonicima, trapezima, elipsama itd.) povezanih strelicama.
- jednim pogledom se može sagledati cela struktura algoritma

Grafički opis

- Dijagrami MIT Scratch/Blockly



- Algoritmi se opisuju slaganjem blokova (poput lego kockica)
- Veći blokovi, koji obuhvataju manje, obično se koriste za opis ponavljanja koraka u manjim blokovima.

Simboli koje koristimo za pojedine algoritamske korake:



početak

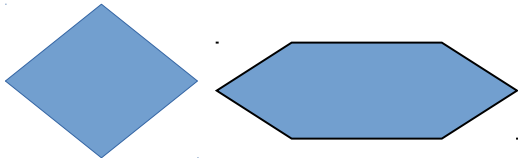
- Definiše početak algoritma
Svaki algoritam započinjemo sa blokom POČETAK



- Definiše ulazne veličine algoritma (unošenje podataka)



- Blok izračunavanja (obrada podataka)



- Grananje u zavisnosti od nekog uslova (odluka)



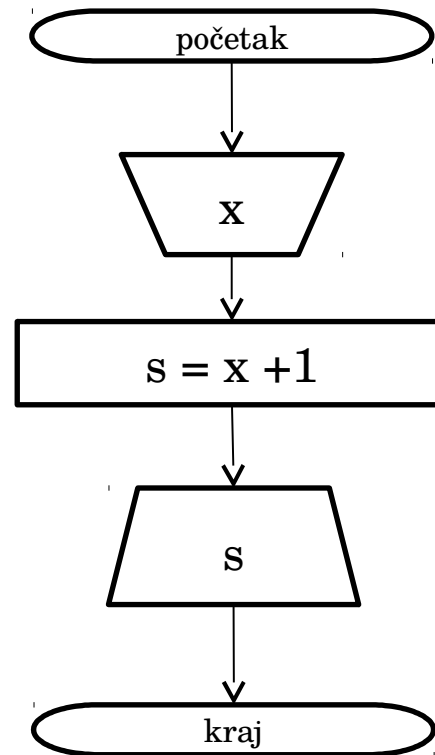
- Definiše izlazne veličine algoritma (izdavanje podataka)



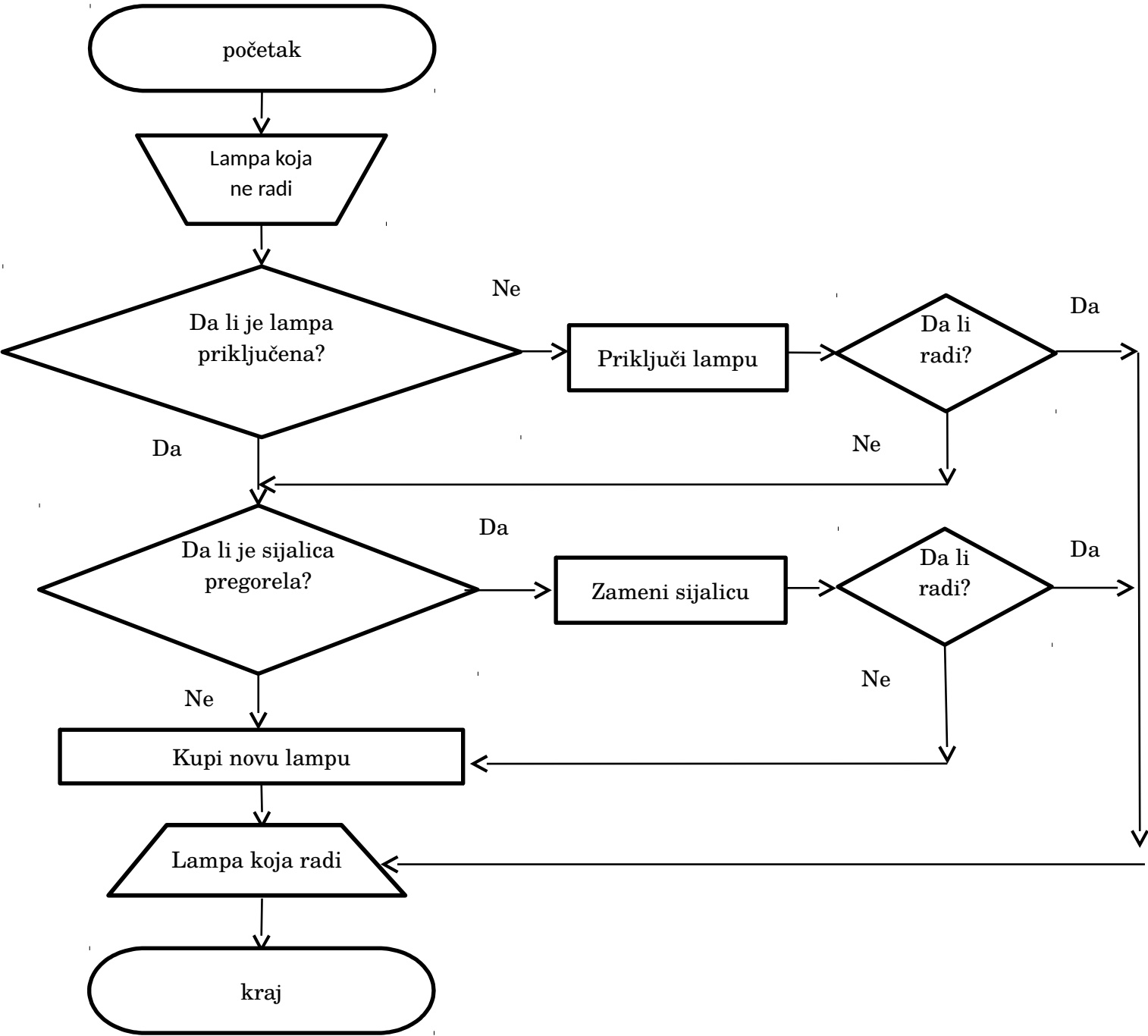
kraj

- Definiše kraj algoritma
Svaki algoritam završavamo sa blokom KRAJ

Primer 1: Algoritam za nalaženje sledbenika nekog broja

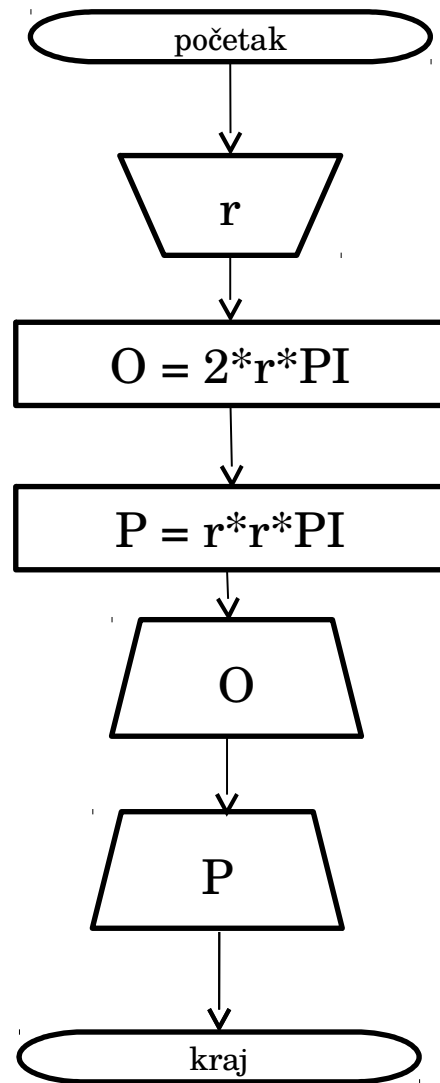


Primer 2: Pokvarena lampa



Primer 3:

Dijagramom toka predstaviti algoritam koji za proizvoljnu vrednost poluprečnika kruga računa i ispisuje njegov obim i površinu



1. Početak

2. Učitaj poluprečnik

3. Izračunaj obim

4. Izračunaj površinu

5. Ispiši obim

6. Ispiši površinu

7. Kraj