Algoritam

Definicija algoritma

- Algoritam je precizna uputa kako obaviti neku radnju ili opis rješenja određenog problema.
- Kroz algoritam cjelokupan zadatak se svodi na rješavanje više jednostavnih, vezanih manjih zadataka.
- Algoritam treba napisati detaljno i jasno, tako da osoba koja nikad nije rješavala postavljeni zadatak može slijedeći korake algoritma doći do rješenja.

Algoritmi u programiranju

- Postoji čitav niz zadataka koji prema uputama algoritma u načelu može riješiti i stroj.
- Računalo zadatak može riješiti samo ako dobije naputak kako to učiniti, i takva naputak nazivamo algoritam.
- Algoritam mora biti napisan na način da se može pretvoriti u naredbe računalnog jezika.

Algoritmi u programiranju

- Algoritam se redovito sastoji od niza relativno jednostavnih operacija čijim se slijednim izvršavanjem može na temelju ulaznih podataka dobiti rezultat.

Struktura algoritma

- Uz svaki algoritam moraju jasno biti definirani početni objekti (ulazni podaci).
- Kao ishod algoritma pojavljuju se završni objekti ili rezultati.
- Algoritam mora biti sastavljen od konačnog broja koraka koji ukazuju na slijed operacija koje treba obaviti nad početnim objektima kako bi se dobili rezultat ili završni objekti.
- Svaki korak algoritma opisan je instrukcijom.

Struktura algoritma

- Izvršenje koraka algoritma naziva se algoritamski proces.
- Algoritam precizno određuje način rješenja nekog problema i jednoznačno određuje što treba napraviti.
- Npr.

  Instrukcija `x=23/0` *neizvediva je jer dijeljenje nulom nije dozvoljeno.*

  Instrukcija `a uvećaj ili smanji za 3` *nije jednoznačna.*

Učinkovitost algoritma

- Učinkovitost:
  - Algoritam je učinkovit ako u konačnom vremenu vodi do rezultata.
- Primjeri:
  - Zbrajanje cijelih brojeva je učinkovito
  - Dijeljenje realnih brojeva nije jer se može pojaviti broj s beskonačno mnogo znamenki, npr. 10/3 = 3.3333333...
    - Algoritam postaje učinkovit tek ako se broj znamenki unaprijed ograniči

Učinkovitost algoritma

- Algoritam je uporabljiv ako se dobije rezultat u konačnom vremenu
- Vrijeme izvođenja mora biti "razumno"

Primjer: Algoritam koji bi izabirao potez igrača šaha tako da ispita sve moguće posljedice poteza, zahtijevao bi milijarde godina na najbržem zamislivom računalu. Zašto?

- 20 mogućih prvih poteza bijelog
- 20 mogućih prvih poteza crnog
- > 20 mogućih drugih poteza bijelog
- > 20 mogućih drugih poteza crnog itd...
- Za 10 poteza svakog igrača, barem 20^{20} kombinacija ~ 10^{26}
- Kad bi se 1 kombinacija analizirala 1 μs, to je 3170979198376 godina!

Primjer algoritma – kiseljenje krastavaca

- Početni objekti:
  - 5 kg krastavaca, 1 l alkoholnog octa (9%), 30 dag šećera, 10 dag soli, kopar, papar
- Krastavce i kopar oprati i posložiti u čiste staklenke
- U 2 l vode dodati ocat, šećer, sol i papar
- Zakuhati uz miješanje
- Vruću otopinu uliti u staklenke
- Staklenke zatvoriti celofanom i gumicom
- Složiti staklenke u široki lonac napunjen vodom do grla staklenki
- Zagrijati vodu do 80 stupnjeva. Ako toplomjer nije raspoloživ, zagrijavati dok se s dna ne počnu dizati mjehurići zraka
- Staklenke izvaditi, obrisati i složiti na police u smočnicu
- Prije kušanja pričekati bar 24 sata
- Završni objekti:
  - kiseli krastavci

Podjela algoritama

- Algoritmi mogu biti:
  - Specijalizirani
    - Mogu se primijeniti samo na pojedine početne objekte (primjer upute za korištenje određenog mobilnog uređaja)
  - Općeniti
    - Dozvoljavaju različite vrijednosti početnih objekata s tim da se kod takvih algoritama definira klasa ulaznih objekata koji su dozvoljeni (primjer zbrajanja dva broja, klasa ulaznih objekata su cijeli brojevi)

Prikaz algoritma

- Osim opisnog načina, kod kojeg je algoritam zapravo opis u svakodnevnom govornom jeziku, računalni algoritam je pogodno prikazati dijagramom tijeka ili pseudokodom.