LABORATORIJSKE VJEŽBE IZ ALGORITAMA I STRUKTURA PODATAKA

Vježba 1

Analiza složenosti algoritma

Analiza složenosti algoritma je važna u računarstvu. Da bi mogli usporediti algoritme, trebamo imati nekakvi kriterij za mjerenje njihove učinkovitosti.

Pretpostavimo da je M algoritam, a n broj ulaznih podataka. Vrijeme i prostor kojeg algoritam koristi dvije su glavne mjere učinkovitosti algoritma. Vrijeme se mjeri brojem **ključnih operacija** – npr. u algoritmima sortiranja i traženja to je broj usporedbi. Za ključne operacije izabiru se one operacije čije izvođenje traje znatno duže od ostalih operacija. Prostor se mjeri maksimalnim memorijskim prostorom potrebnim za izvođenje algoritma.

ANALIZA A PRIORI

A priori analiza daje trajanje izvođenja algoritma kao vrijednost funkcije nekih relevantnih argumenata. Koristi se O-notacija :

$f(n) = O(g(n))$, ako i samo ako postoje dvije pozitivne konstante c i n_0 takve da vrijedi $|f(n)| \le c|g(n)|$ za sve $n > n_0$. Traži se najmanji $g(n)$ za koji to vrijedi.

A priori analizom dobije se vrijeme izvođenja algoritma $O(g(n))$.

Za dovoljno velik n vrijedi:

$$O(1) < O(\log n) < O(n) < O(n \log n) < O(n^2) < i(n^3) < ... < O(2^n)$$

Definirana je i donja granica i asimptotsko vrijeme izvođenja algoritma :

Donja granica za vrijeme izvođenja algoritma $f(n) = \Omega(g(n))$, ako i samo postoje dvije pozitivne konstante c i n_0 takve da vrijedi $|f(n)| \ge c|g(n)|$ za sve $n > n_0$.

Asimptotsko vrijeme izvođenja je $f(n) \sim o(g(n))$ ako je :

$$\lim_{n \to \infty} f(n) / g(n) = 1$$

ANALIZA A POSTERIORI

Analizom a posteriori eksperimentalno se određuje vrijeme potrebno za izvođenje algoritma na konkretnom računalu.
Jedan od najjednostavnijih načina mjerenja vremena u C/C++-u je slijedeći:

```
#include <time.h>
...
time_t t1, t2;
...
t1 = clock();
...
// ovdje dolazi dio programa za koji mjerimo vrijeme
...
t2 = clock();
printf( "Vrijeme trajanja je %dms\n",t2-t1 );
```

Kao što vidimo iz navedenog programskog odsječka koristi se gotova funkcija clock() koja daje vrijeme od pokretanja računala do trenutka poziva u milisekundama. Na kraju se uzima razlika dva vremena koja je isto tako u milisekundama. Navedeni primjer je namijenjen za konzolne aplikacije pod Windows okruženjem, dok se pod UNIX-om treba raditi malo drugačije.
U drugim programskim jezicima mjerenje vremena radi se na vrlo sličan način.

PRIPREMA ZA VJEŽBU :

Uporabom funkcije `int rand(void);` koja je definirana u `<stdlib.h>` načiniti općenitu funkciju za generiranje polja s `n` pseudoslučajnih brojeva čija se vrijednost nalazi između zadane donje i gornje granice. Prototip funkcije je:
`void gen_arr( float V[], int n, float dg, float gg );`

Načiniti slijedeće funkcije:

a) sekvencijalno pretraživanje
   `int sekv_pret( float V[], int n, float x );`
   Funkcija vraća -1 ako se traženi broj x ne nalazi u V, u suprotnom vraća prvo mjesto u nizu na kojem se nalazi x.
b) Sortiranje
   `void sort( float V[], int n );`
   Upotrijebiti bilo koji algoritam za sortiranje niza V. Napraviti uzlazno sortiranje.
c) binarno pretraživanje
   `int bin_pret( float V[], int n, float x );`
   Ovdje V mora biti uzlazno sortirani niz. Funkcija vraća -1 ako se traženi broj x ne nalazi u V, u suprotnom vraća prvo mjesto u nizu na kojem se nalazi x.

Korisnik programa sam određuje broj članova polja n.

Koje je vrijeme izvođenja $O(g(n))$ svake od tri navedene funkcije?

LABORATORIJSKA VJEŽBA 1 - ALGORITMI I STRUKTURE PODATAKA

Prezime i ime: _____

Broj indeksa: _____

A) Koja je teorijska vremenska složenost svakog od navedenih algoritama?

_____

_____

_____

B) Popuniti tablicu tako da napravite i pokrenete programe (unijeti vrijeme u milisekundama, prilikom testiranja programa za pretraživanje, zadati vrijednost X koje nema u nizu, tako da algoritam traje najduže):

Broj elemenata n	**Sekvencijano Pretraživanje**	**Sortiranje**	**Binarno pretraživanje**
10.000			
20.000			
50.000			
100.000			
200.000			
500.000			
1.000.000			
2.000.000			
5.000.000			
10.000.000			

C) Skicirajte vrijeme izvršavanja u odnosu na broj elemenata niza na grafu:

D) Napišite vlastiti komentar

_____

_____

_____

_____