Oblikovanje i analiza algoritama (akad. god. 2009./10.)

Domaće zadaće

Upute ili "pravila igre":

- Za zadaću se bira 1 (**jedan**) od ponuđenih 5 zadataka. Kao **bonus**, dozvoljeno je riješiti i više zadataka.
- Rješenje je program s popratnom dokumentacijom.
- Zadnji rok za predaju zadaće je **ponedjeljak, 1. veljače 2010., do 14 sati**.

Kodeks ponašanja: Ista ili vrlo slična rješenja se **poništavaju**.

Napomena. Programski zadaci su **namjerno** sažeto formulirani, tako da propisuju samo nužni dio posla. Sve ostalo: planiranje ulaza, izlaza, izbora raznih algoritama, implementacije, testiranja i sl., je "slobodno" i nagrađuje se kao dio rješenja. Posebno, probajte napraviti što efikasnije implementacije odgovarajućih algoritama. Za rješenje smijete koristiti i dodatnu literaturu, uključivo i programe s weba ("sve što nađete"), samo propisno citirajte. Ako koristite "tuđe" algoritme, budite oprezni i pogledajte/testirajte ih prije korištenja!

Primjer jednostavnog mjerenja vremena (`dsecnd`) za Hanojske tornjeve je dostupan na webu. Smijete koristiti i bolju štopericu, ako ju nađete (molim, pošaljite mi ju, za buduće generacije).

Zadatak 1. Mergesort algoritam za sortiranje polja ili liste.

- Napišite program koji uspoređuje razne varijante Mergesort algoritma za sortiranje polja, odnosno, vezane liste cijelih brojeva. Kod sortiranja polja, dozvoljeno je koristiti jedno dodatno polje. Kod sortiranja liste, dozvoljene su samo zamjene pokazivača. Razne varijante algoritma sastoje se u tome da polje/listu dijelimo na k dijelova podjednake duljine, s tim da k varira od 2 (standardni algoritam), pa sve do $\sqrt{n}$, gdje je n broj elemenata u polju/listi. Testirajte program za razne vrste polja/liste, variranjem duljine polja/liste i rasporeda elemenata u polju/listi. Smijete usporediti iste varijante algoritma na raznim strukturama: na polju i na listi.

Literatura: predavanja, predavanja iz Programiranja 2, dijelovi 5. poglavlja iz JS knjige (dostupno na webu), Knuth 3 (javite se, ako želite), potražite po webu.

Zadatak 2. Minimalno razapinjuće stablo.

- Napišite program koji implementira Kruskalov algoritam za nalaženje minimalnog razapinjućeg stabla zadanog neusmjerenog grafa $G = (V, E)$ s pozitivnim težinama bridova. Iskoristite strukturu disjunktnih skupova. Smijete usporediti i razne varijante za pojedine operacije u toj strukturi (pogledajte predavanja). Testirajte program za razne vrste grafova, variranjem broja vrhova i broja bridova ("gustoća") grafa.

Literatura: predavanja, dijelovi 7. i 3. poglavlja iz knjige (dostupno na webu).

Zadatak 3. Brzo množenje velikih brojeva.

- Napišite program koji uspoređuje trajanje standardnog i nekih "brzih" algoritama za množenje velikih prirodnih brojeva. Možete koristiti Karacubin algoritam i generalizacije iz familije brzih algoritama s parametrom r, za male vrijednosti r. Obratite pažnju na realizaciju rekurzivnog dijela algoritma, tako da alokacija/dealokacija memorije ne troši previše vremena. Pokušajte izbjeći nepotrebna kopiranja blokova znamenki! Testirajte program za razne "duljine" brojeva.

Literatura: predavanja, potražite po webu.

Zadatak 4. Brzo množenje kvadratnih matrica.

- Napišite program koji uspoređuje trajanje standardnog i nekih "brzih" algoritama za množenje kvadratnih matrica realnih brojeva (double). Možete koristiti rekurzivni Strassenov algoritam i Vinogradovu modifikaciju tog algoritma s predavanja, složenosti $O(n^{\log_2 7})$. Obratite pažnju na realizaciju rekurzivnog dijela algoritma, tako da alokacija/dealokacija memorije ne troši previše vremena. Pokušajte izbjeći nepotrebna kopiranja matrica! Smijete testirate i razne "blokovske" realizacije bilo kojeg osnovnog algoritma, s idejom efikasnog korištenja cache memorije. Testirajte program za razne redove matrica (barem do $n = 1000$).

Literatura: predavanja, potražite po webu.

Zadatak 5. Brzo množenje polinoma.

- Napišite program koji uspoređuje trajanje standardnog i nekih "brzih" algoritama za množenje polinoma s realnim ili kompleksnim koeficijentima. Možete koristiti razne implementacije brze Fourierove transformacije, složenosti $O(n \log n)$ (bar kad je n potencija od 2). Obratite pažnju na realizaciju rekurzivnog dijela algoritma, tako da alokacija/dealokacija memorije ne troši previše vremena. Pokušajte izbjeći nepotrebna kopiranja vektora koeficijenata!

Literatura: predavanja, H. S. Wilf: "Algorithms and Complexity" (javite se, ako želite), potražite po webu.