

Oblikovanje i analiza algoritama (akad. god. 2008./09.)

Domaće zadaće

Upute ili “pravila igre”:

- Za zadaću se bira 1 (**jedan**) od ponuđena 4 zadatka. Kao **bonus**, dozvoljeno je riješiti i više zadataka.
- Rješenje je program s popratnom dokumentacijom.
- Zadnji rok za predaju zadaće je **ponedjeljak, 16. veljače 2009., 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 je nađete (molim, pošaljite mi ju, za buduće generacije).

Zadatak 1. Sortiranje polja.

- Napišite program koji uspoređuje trajanje Quicksort i raznih drugih algoritama za sortiranje polja cijelih brojeva, poput Mergesort, Heapsort, Shellsort, Radixsort i slično. Testirajte program za razne vrste polja, variranjem duljine polja i rasporeda elemenata u polju. Za test, smijete usporediti trajanje izabranih algoritama i Quicksorta iz standardne C-biblioteke (ako koristite C).

Literatura: predavanja, dijelovi 3. i 6. poglavlja iz JS knjige (dostupno na webu), Knuth 3 (javite se, ako želite), potražite po webu.

Zadatak 2. 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 3. 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 testirati i razne “blokove” 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 4. 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.