# Oblikovanje i analiza algoritama (akad. god. 2010./11.)

## Domaće zadaće

**Upute ili "pravila igre":**

- Za zadaću se bira 1 (**jedan**) od ponuđenih 5 zadataka.
- Rješenje je program koji radi to što treba, s popratnom dokumentacijom (ne mora je biti, ali dozvoljeno je, na primjer, sažeto prikazati rezultate opširnijeg testiranja).
- Zadnji rok za predaju zadaće je **petak, 28. siječnja 2011., do 16 sati**.

**Kodeks ponašanja:** Tuđa rješenja (autor programa je netko drugi), 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).

**Uputa** za izravne linkove na literaturu. Početak (prefiks) pune adrese za **sve** članke je

`http://degiorgi.math.hr/oaa/`

Na to treba dodati, tj. konkatenirati relativnu adresu (sufiks) za svaki pojedini članak. Tako dobivenu cijelu adresu treba kopirati u web preglednik i onda spremiti članak.

**Zadatak 1**. Hanojski tornjevi s više igala.

- Promatramo problem Hanojskih tornjeva s $n$ diskova i $k \geq 3$ igala, tj. imamo $k-2$ pomoćne igle. Napišite program koji nalazi rješenje ovog problema za razne vrijednosti $n$ i $k$, u što je moguće manjem broju poteza. Dovoljno je naći samo broj poteza u rješenju, a ne i njihov redoslijed. Dozvoljeno je koristiti razne tehnike tzv. ekstenzivnog pretraživanja prostora poteza, kao i algoritme na bazi dinamičkog programiranja, poput Frame–Stewartove rekurzije, iako nije dokazano da daju najmanji mogući broj poteza. Smijete usporediti i razne algoritme za isti problem. Testirajte program za razne vrijednosti $n$ i $k$, u nekim razumnim granicama trajanja. Pripazite na raspon prikazivih cijelih brojeva za broj poteza.

Literatura: predavanja (kao uvod, za $k = 3$), članak (292 kB) na sljedećoj relativnoj adresi

`oaa_lit/hanoi_k_majumdar.pdf`

članak M. Randa (link na webu), potražite po webu.

**Zadatak 2**. Shellsort algoritam za sortiranje niza (polja).

- Napišite program koji uspoređuje razne varijante Shellsort algoritma za sortiranje niza (polja) cijelih brojeva. Osnovni algoritam koristi sortiranje umetanjem (insertion sort) za sortiranje podnizova, a podnizovi su određeni tzv. nizom inkremenata $g_i$ (ili $h_i$), koji se koristi u **padajućem** poretku, ovisno o duljini polja. Razne varijante algoritma, ovisno o izboru ovog niza inkremenata, mogu imati bitno različitu složenost. Testirajte program za razne nizove inkremenata i razne vrste polja, variranjem duljine polja i rasporeda elemenata u polju. Za sortiranje podnizova smijete koristiti i druge "jednostavne" algoritme sortiranja, a ne samo "insertion sort".

Literatura: zadaci 6.14–6.16 na kraju 6. poglavlja iz JS knjige (dostupno na webu), Knuth 3 (javite se, ako želite), članak R. Sedgewicka (link na webu), potražite po 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.