

Oblikovanje i analiza algoritama (akad. god. 2014./2015.)

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 **utorak, 17. veljače 2015., do 14 sati**.

Kodeks ponašanja: Ako **ne razumijete** što vaš program radi, rješenje se **poništava!** Tuđa rješenja (autor programa je netko drugi), ista ili vrlo slična rješenja se **poništavaju**. To se može dogoditi i naknadno, tj. retroaktivno, ako se kasnije pojave “kopije”!

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, morate znati što oni rade. Dodatno, 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. konkatenerati relativnu adresu (sufiks) za svaki pojedini članak. Tako dobivenu cijelu adresu treba kopirati u web preglednik i onda spremiti članak.

Zadatak 1. Brzo računanje n -tog člana rekurzije.

- Napišite program za brzo računanje n -tog člana niza zadanog k -članom homogenom rekurzijom s konstantnim koeficijentima. Uzmite da su koeficijenti u rekurziji i početni članovi nenegativni cijeli brojevi, tako da isto vrijedi i za sve članove niza. Program treba egzaktno izračunati n -ti član niza. Implementirajte operacije zbrajanja i množenja velikih brojeva u nekoj povoljno izabranoj bazi b (poželjno da je potencija od 10, za lakši ispis). Smijete koristiti i algoritme za brzo množenje velikih brojeva. Testirajte za rekurzije poput Fibonaccijevih brojeva, s relativno malim brojem k (ne treba preko 5) i dovoljno velikim n . Poželjno je da program radi barem za n u tisućama! Obratite pažnju na programsku realizaciju “rekurzivnih” dijelova algoritma, tako da eventualna alokacija/dealokacija memorije ne troši previše vremena. Posebno, pokušajte izbjeći nepotrebna kopiranja polja (nizova znamenki i slično)!

Literatura: predavanja, potražite po webu.

Zadatak 2. Pretraživanje teksta, približno traženje.

- U zadanom tekstu (može biti zadan kao dugačka tekst-datoteka) treba pronaći zadani uzorak (opet, može biti zadan kao dugačka tekst-datoteka). Napišite program koji pronalazi prvu (ili sve) pojave zadanog uzorka u tekstu. Smijete iskoristiti i usporediti razne algoritme za pretraživanje teksta. Testirajte program za razne vrste teksta i uzorka, obzirom na sadržaj — na primjer, obični tekst ili ograničen samo “binarni” tekst (niz nula i jedinica), kao i za razne duljine teksta i uzorka. Kao dodatak ili alternativno, smijete implementirati i algoritme za približno prepoznavanje zadanog uzorka u tekstu.

Literatura: 9. poglavlje iz JS knjige (JS 367–428, dostupno na webu), potražite po webu.

Zadatak 3. Najdulji zajednički podniz.

- Zadana su dva niza znakova (oba mogu biti zadana kao dugačke tekst-datoteke). Napišite program koji nalazi **najdulji** zajednički podniz u ta dva niza. Dozvoljeno je usporediti razne algoritme za rješenje ovog problema. Testirajte program za razne duljine nizova, kao i za razne vrste nizova obzirom na osnovni alfabet znakova — na primjer, obični tekst ili ograničen samo “binarni” tekst (niz nula i jedinica).

Literatura: dijelovi poglavlja o dinamičkom programiranju u knjigama JS i ALS (JS 342–349, ALS 205–208, dostupno na webu), potražite po webu.

Zadatak 4. Hanojski tornjevi s više igala, ograničeni potezi.

- Promatramo problem Hanojskih tornjeva s n diskova i $k \geq 3$ igala, tj. imamo $k - 2$ pomoćne igle. Dodatno ograničenje na poteze (tzv. “restricted move”) je da pojedini disk smijemo prebaciti prebaciti samo na **susjednu** iglu (kao u 1. kolokviju iz 2012. godine). 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 redosljed. Smijete usporediti i razne algoritme za isti problem. Ako ne znate kako riješiti problem s ograničenim potezima, riješite problem **bez** ograničenja. 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 $k = 3$), potražite po webu. Za opće poteze: članak (292 kB) na sljedećoj relativnoj adresi

`oaa_lit/hanoi_k_majumdar.pdf`

i članak M. Randa (link na webu).

Zadatak 5. 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. Također, smijete usporediti trajanje ovih algoritama s trajanjem `qsort` funkcije iz standardne C biblioteke.

Literatura: predavanja, predavanja iz Programiranja 2, dijelovi 5. poglavlja iz JS knjige (JS 219–225, dostupno na webu), CLRS 29–35, Knuth TAOCP 3 (javite se, ako želite), potražite po webu.