

Strukture podataka

I Kolokvijum, 06.04.2012

1. (20 poena)
 - a. (5 poena) Da li izbor strukture podataka za rešavanje konkretnog problema utiče na efikasnost implementacije u nekom programskom jeziku? Ukoliko imate fiksni broj elemenata koje treba da čuvate, da li ćete da izaberete statičku ili dinamičku memorijsku reprezentaciju izabrane strukture?
 - b. (15 poena) Koristeći pseudojezik napisati funkciju *Prikaži* koja iz sortiranog jednodimenzionalnog polja A sa N elemenata traži element koji je jednak prosečnoj vrednosti svih elemenata polja. Funkcija treba da vrati poziciju takvog elementa ili NULL ako ta vrednost ne postoji u polju.
2. (30 poena)
 - a. (10 poena) Dati su magacin M koji je lančano memorisan, i red Q koji je sekvencijalno memorisan. Magacin je inicijalno prazan, a u redu su elementi D i E . Niz elemenata A , B i C upisati navedenim redom u magacin M . Nakon toga četiri puta pročitati (obrisati) element iz magacina M i svaki pročitani element upisati u red Q . Nakon toga pročitati dva puta elemente iz reda Q i upisati ih u magacin M . Prikazati inicijalno stanje magacina M i reda Q , kao i stanja nakon svake operacije.
 - b. (10 poena) Koristeći pseudojezik napisati proceduru *Pop* za čitanje elementa iz magacina M koji je lančano memorisan.
 - c. (10 poena) Koristeći pseudojezik napisati proceduru *Dequeue* za brisanje elementa iz reda Q koji je sekvencijalno memorisan.
3. (25 poena) Projektovati klasu za rad sa strukturom LRU (*least recently used*) keša. Struktura treba da omogućiti smeštanje maksimalno N elemenata tipa *Node*. Prilikom dodavanja novog elementa, ukoliko je struktura već puna, potrebno je obrisati element kojem se najdavnije pristupalo. Napisati:
 - a. (5 poena) Deklaraciju klase za rad sa ovom strukturom,
 - b. (10 poena) C++ funkciju za čitanje elementa sa zadatim ključem. Smatrati da struktura *Node* sadrži ključno polje.
 - c. (10 poena) C++ funkciju za dodavanje novog elementa u strukturu.
4. (25 poena) Za zadatu Hash tablicu se za rešavanje kolizije koristi unutrašnje ulančavanje sinonima. Tablica treba da služi za smeštanje informacija o studentima na Univerzitetu. O svakom studentu pamte se: broj indeksa, ime, prezime i godina studija. Procenjeni broj studenata na Univerzitetu je 25000.
 - a. (5 poena) Odrediti veličinu tablice, predložiti ključ i napisati šta je sve neophodno za implementaciju.
 - b. (5 poena) Napisati deklaraciju klase,
 - c. (15 poena) Ako se za primarnu transformaciju koristi hash funkcija $h(k)$, gde je k ključ koji ste izabrali, napisati C++ funkciju za brisanje podataka iz tablice o zadatom studentu. Podrazumevati da postoji funkcija *VratiULRMP(p)* koja "vraća" obrisani element p u listu raspoloživog memorijskog prostora unutar tablice.

Napomena:

Ukupan broj poena je 100.

Ovaj Kolokvijum se smatra uspešno položenim ako student osvoji najmanje 50 poena.