

Strukture podataka

Jun 2 2020

1. (25 poena) Polje, Niz, Lančane liste

- a. (12 poena) Ako su dostupne osnovne operacije za obradu reči (izdvajanje podniza, indeksiranje, konkatencija i nalaženje dužina niza), napisati pseudokod operacije koja predstavlja modifikaciju kompozitne operacije za umetanje u tekst T , tako što se umesto na poziciju k , novi niz S dodaje na poziciju iza zadatog uzorka P (ako P postoji u tekstu T).
- b. (13 poena) Napisati metodu članicu klase jednostruko lančane liste **void Exchange(SLList& list, int value, int length)** koja zamenjuje delove dužine **length** dve lančane liste jednake dužine (prva lista je objekat za koji je metoda pozvana, dok je druga prosleđena kao argument). Zamenjuje se deo iz prve liste od pozicije pos na kojoj se nalazi vrednost **value** dužine **length** sa delom druge liste od iste pozicije pos (pronađene u prvoj listi) dužine **length**. Smatrati da delovi koji treba da se zamene postoje. Voditi računa o efikasnosti rešenja.

2. (25 poena) Magacin, Red, Heš tablice

- a. (13 poena) Opisati Heš tablicu i proces traženja elementa iz Heš tablice ako se kao primarna transformacija koristi heš funkcija $h(k)$, a kao sekundarna spoljašnje ulančavanje sinonima.
- b. (12 poena) Retko posednuta matrica zapamćena je u rasutoj tablici (heš tablica sa otvorenim adresiranjem). Napisati funkcije za: primarnu transformaciju, sekundarnu transformaciju, umetanje elementa, čitanje elementa i množenje dve retko posednute matrice. Smatrati da je tablica dimenzija $m \cdot n$, i da ima manje od 20% nenulatih elemenata tipa *double*. Samo nenulti elementi se pamte u tablici. **Napomena:** Implementaciju prilagoditi konkretnom problemu. Generičke funkcije neće biti bodovane.

3. (25 poena) Stabla

- a. (13 poena) Dat je niz $A=(2,13,1,12,7,15,18)$. Od niza A formirati stablo binarnog traženja S i Heap H (prikazati postupak korak po korak!). Prikazati rezultat primene postorder obilaska stabla S i stabla H .
- b. (12 poena) Za neuređeno dinamički implementirano binarno stablo, napisati funkciju **BSTNode* findClosestValueLeaf(BSTNode* root, int value)** koja određuje i vraća list čija je vrednost ključa najbliža prosleđenoj vrednosti **value**. Rešenja koja koriste statičke i globalne promenljive će se ocenjivati sa najviše 30% poena.

4. (25 poena) Grafovi

- a. (12 poena) Za lančanu reprezentaciju orijentisanog grafa, napisati pseudokod operacije koja određuje da li je čvor u jedan od suseda čvora v . Za čvorove u i v su date njihove vrednosti.
- b. (13 poena) Za dinamički usmereni graf implementiran listom susedstva napisati funkciju **int findLeastNodesPath(Node* first, Node* last)** koja vraća broj čvorova onog puta između čvorova **first** i **last** koji prolazi kroz najmanji broj ostalih čvorova grafa.

PREDMETNI NASTAVNIK