

Strukture podataka

Septembar 2019

1. (25 poena) Magacin, Red, Lančane liste

- a. (13 poena) Zadat je tekst T , uzorak P i uzorak Q . Napisati pseudokod operacije $ZAMENA(T, P, Q)$ kojim se vrši zamena mesta uzorka P i uzorka Q u tekstu T . Podrazumevati da postoje osnovne operacije za rad sa nizovima (*Substring*, *Index*, *Concat*, *Length*), a za svaku od njih koju koristite u pseudokodu napišite pored naziva parametre i kratak opis šta radi.

Primer: $ZAMENA("Petar Petrović Milan Aleksandar Zoran Jovanović", "Petrović", "Jovanović") \rightarrow$
 $\rightarrow "Petar Jovanović Milan Aleksandar Zoran Petrović"$

- b. (12 poena) Napisati funkciju **bool DeleteElement(int i, int j)**, koja briše element sa indeksima i i j u lančanoj reprezentaciji retko posednute matrice. Ako je brisanje uspešno, funkcija vraća *true*, u protivnom vraća *false*.

2. (25 poena) Heš tablice

- a. (12 poena) Opisati opšti postupak kod brisanja podatka sa ključem K iz Heš tablice. Kakve se promene zahtevaju kod ovog postupka ako se koristi spoljašnje ulančavanje sinonima?

- b. (13 poena) Aplikacija za parking servis beleži obidena vozila koja je operater uočio obilazeći svoj sektor. Za svaki automobil se pamti njegova tablica, vreme kada je automobil uočen (celi broj Unix time) i da li je parking plaćen. Radi brzog pribavljanja podataka o uočenim vozilima, podaci se smeštaju u Heš tablicu implementiranu korišćenjem otvorenog adresiranja sa kvadratnim traženjem. Format tablice je SS-BBBB-SS, pri čemu S označava slovo a B označava broj. Implementirati heš funkciju koja minimizira sinonime i vrši dobro rasipanje u tablici za izabrani ključ i implementirati funkcije **bool isVisited(char* key)**, **bool hasPaid(char* key)**, **updatePaid(char* key, bool paid)** i **visit(char* key, char* time, bool paid)**.

3. (25 poena) Stabla

- a. (13 poena) Napisati pseudokod operacije za traženje čvora u stablu binarnog traženja S na osnovu zadate vrednosti tog čvora. Koristiti sekvencijalnu reprezentaciju stabla. Prikazati stablo binarnog traženja koje se dobija na osnovu elemenata niza $A=(2,13,18,1,7)$ i prikazati njegovu sekvencijalnu memorijsku reprezentaciju.

- b. (12 poena) Napisati funkciju **BSTNode* deepestLeaf(BSTNode* root)** koja u dinamičkom binarnom stablu određuje čvor koji je list i koji se nalazi na najvećoj dubini u stablu. Ukoliko i levo i desno podstablo imaju listove na istoj dubini, izabrati list iz levog podstabla. **Napomena:** Nije dozvoljeno koristiti druge metode implementacije stabla. Dozvoljeno je koristiti druge funkcije sa porizvoljnom listom argumenata. Voditi računa o optimalnosti rešenja.

4. (25 poena) Grafovi

- a. (13 poena) Zadat je orijentisani graf G listom čvorova $\{A,B,C,D,E,F\}$ i listom grana (gde je svaki element u navedenoj listi: (čvor, čvor, težina potega)) $\{(B,C,1), (A,C,3), (A,D,3), (C,E,5), (D,E,7), (C,D,1), (E,D,4), (F,F,2)\}$.

i. Prikazati lančanu reprezentaciju grafa G , koja u čvorovima za potege sadrži i podatak o težini.

ii. Za graf G predstavljen ovakvom lančanom reprezentacijom, napisati pseudokod operacije ažuriranja vrednosti težina na vrednost V , za sve one potege koji počinju u čvoru čiji je info deo C .

- b. (12 poena) Dat je orijentisan graf predstavljen listom suseda. Napisati funkciju **Node* FindMaxReachableWithRoundTrip()**, koja određuje čvor X u orijentisanom grafu od koga se može stići do najvećeg broja čvorova u grafu tako da za svaki put od čvora X do čvora Y važi da se istim putem samo u suprotnom smeru može stići iz čvora Y u čvor X . **Napomena:** Voditi računa o efikasnosti napisane funkcije. Smatrati da ne postoje implementirane pomoćne funkcije za rad sa grafom, ali da postoje implementirane pomoćne strukture (ukoliko su potrebne).