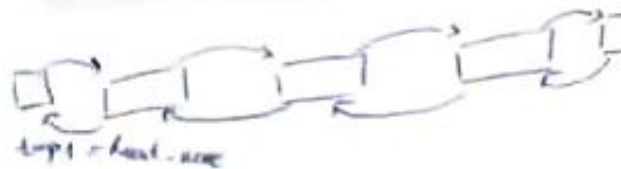


while(temp != null)
{ if (temp->N > temp->W)
temp = temp->W
temp = temp->N }



Strukture podataka

Oktober 2018

temp = head -> next

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

- a. (12 poena) Napisati pseudokod operacije koja u nizu A od N elemenata nalazi sve elemente koji su veći od srednje vrednosti svih elemenata niza.
- b. (13 poena) Napisati funkciju void UmetniListu(DLList uredjena, DLList nova) koja elemente neuređene dvostruko ulančane liste nova umeće u dvostruko ulančanu listu uredjena tako da ova lista ostane uređena u rastućem redosledu. Prilikom dodavanja svakog sledećeg elementa iz liste nova traženje mesta započeti od pozicije na kojoj se nalazi prethodno umetnuti element u listi uredjena.

2. (25 poena) Heš tablice

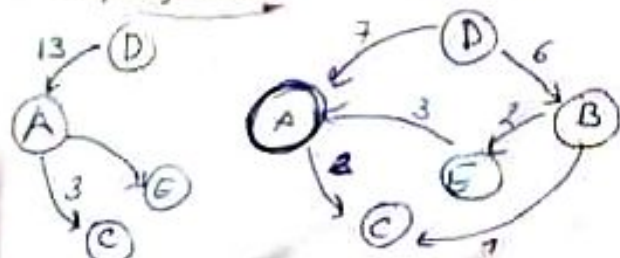
- a. (12 poena) Opisati opšti postupak kod brisanja podatka sa ključem K iz Heš tablice. Kakve se promene zahtevaju kod ovog postupka ako se koristi spoljašnje ulančavanje sinonima?
- b. (13 poena) Radi brzog pristupa podacima o mobilnim telefonima, softver bazne stanice mobilne telefonije čuva podatke o vremenu poslednjeg pristupa mobilnog telefona u heš tablici sa otvorenim adresiranjem. Implementirati na programskom jeziku C++ funkciju heš tablice bool insertOrUpdateCellData(c), koja smešta ili osvežava podatke o telefonu koji pristupa baznoj stanici mobilne telefonije. CellData klasa sadrži podatke o broju mobilnog telefona, IMEI broju (15 dekadnih cifara) i trenutku pristupa (smatrati da je implementiran kao jedan integer - Unix timestamp). Odrediti najpogodniji podatak koji će biti ključ, imajući u vidu da jedan telefonski broj može pristupiti baznoj stanici sa više različitih telefona. Predložiti i pogodnu primarnu i sekundarnu heš funkciju. Smatrati da mreža mobilne telefonije ima najviše 1000000 pretplatnika.

3. (25 poena) Stabla

- a. (12 poena) Dat je niz A=(2,13,18,12,7,15,1). Od niza A formirati stablo binarnog traženja S i Heap H. Prikazati rezultat primene postorder obilaska stabla S i stabla H.
- b. (13 poena) Napisati funkciju double CalcExpression(Node* node), koja izračunava vrednost aritmetičkog izraza zadatog preko dinamički implementiranog binarnog stabla. Neterminalni čvorovi sadrže operacije +, -, *, / i %. Terminalni realne vrednosti dvostruke tačnosti. Prilikom implementacije obavezno navesti deklaraciju klase Node.

4. (25 poena) Grafovi

- a. (13 poena) Zadati 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)}. Navesti redosled obilaska grafa po dubini, kao i sadržaj korišćene pomoćne strukture u svakom koraku, ako obilazak počinje od čvora B. Za graf G predstavljen lančanom reprezentacijom, koja u čvorovima za potege sadrži i podatak o težini, 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) Napisati funkciju void fuseNodes(Node* a, Node* b), koja vrši fuziju dva zadata čvora, a i b, dinamički implementiranog težinskog usmerenog grafa. Voditi računa da se težine grana koje izviru ili uviru u isti čvor prilikom fuzije sabiraju. Smatrati da nakon fuzije čvorovi a i b postaju čvor a.



PREDMETNI NASTAVNIK