

Algoritmi i strukture podataka

ponovljeni završni ispit

4.7.2006.

Napomene za sve zadatke:

Nije dopušteno korištenje goto naredbe, te statičkih i globalnih varijabli.

U svakom zadatku u kojem se koristi struktura, strukturu treba i definirati.

1. U jednostruko povezanu listu spremaju se cjelobrojni podaci(**long**). Napisati funkciju koja će dvije takve **uzlazno sortirane** liste spojiti u jednu koja sadrži sve elemente obje liste i koja je **silazno sortirana**. Funkcija mora vratiti pokazivač na početak nove liste(glave zadanih lista treba postaviti na NULL). Obavezno iskoristiti činjenicu da su zadane liste sortirane(funkcija mora imati složenost **O(n)**). Funkcija mora imati prototip:

zapis *merge(zapis **glava1, zapis **glava2);

Napomena: Memoriju nije potrebno zauzimati i oslobađati. Dovoljno je novu listu napraviti od postojećih elemenata u zadanim listama. Može se pretpostaviti da zadane liste nemaju istih elemenata.

2. U binarno stablo spremaju se cjelobrojni podaci (**int**). Zadan je niz brojeva: **2, 7, 3, 1, 4, 10, 8, 9, 6 i 5**.

a) Ilustrirati stvaranje sortiranog binarnog stabla (lijevo manji desno veći) od zadanog niza brojeva ako se bojevi u stablu ubacuju redosljedom kojim su gore navedeni. Nacrtati izgled stabla nakon dodavanja svakog elementa.

b) Napisati funkcije za preorder, inorder, i postorder ispis stabla. Obavezno navesti koja funkcija koristi koji način obilaska stabla.

c) Napisati što bi svaka od funkcija iz **b)** ispisala za stablo iz **a)**.

3. U objektno realizirani stog i red (razredi **Stog** i **Red**) spremaju se cjelobrojni podaci (long). Za oba razreda definirane su samo funkcije za stavljanje i skidanje jednog elementa(vraćaju 1 ako su uspješno obavile posao, a 0 inače), te prazan konstruktor.

Stog::Stog();

Red::Red();

int Stog::Stavi(long element);

int Red::Stavi(long element);

int Stog::Skini(long *element);

int Red::Skini(long *element);

Napisati funkciju koja će sve elemente iz zadanog reda prepisati u zadani stog, ali tako da će ti elementi iz stoga izlaziti istim redoslijedom kojim bi izlazili iz reda. Zadani red na kraju funkcije mora ostati nepromijenjen. Funkcija mora vratiti broj prebačenih elemenata i mora imati prototip:

int prebaci(Stog *zadaniS, Red *zadaniR);

Napomena: Dozvoljeno je korištenje pomoćnog stoga i reda. Razrede Stog i Red ne treba pisati.

4. Zadan je niz brojeva: **2, 7, 3, 1, 4, 10, 8, 9, 6 i 5.**

a) Ilustrirati stvaranje gomile od zadanog niza brojeva postupkom koji za najgori slučaj ima složenost **$O(n)$** . Ispisati sadržaj polja ili stabla nakon svake zamjene dvaju elemenata.

b) Ilustrirati postupak heapsort(počevši od gomile stvorene u **a**) dijelu zadatka). Obavezno ispisati sadržaj polja (**NE STABLA**) nakon svake zamjene dvaju elemenata.

Svaki zadatak je nosio 7.5 bodova.