Univerzitet u Tuzli

Fakultet Elektrotehnike

Uvod u Računarske Algoritme

Zadaća 2

Algoritmi sortiranja. Divide and conquer algoritmi

Tuzla, 11. 5. 2020.

Napomena

U svim problemima koji slijede nije dozvoljena upotreba komandi i funkcija koje dosad nisu korištene na predavanjima ili vježbama. Smije se koristiti std::sort ukoliko je potrebno. Dozvoljena je upotreba kontejnera iz standardne biblioteke (std::vector i std::list), kao i C nizova. Nerekurzivna rješenja se ne mogu smatrati tačnim ukoliko je u zadatku naglašeno da je potrebno koristiti rekurziju.

Zadatak 1

Implementirati algoritam sortiranja bubble sort sa potpisom:

```
template <typename It>
void bubblesort(It begin, It end)
```

Zadatak 2

Implementirati algoritam sortiranja selection sort sa potpisom:

```
template <typename It>
void selectionsort(It begin, It end)
```

Zadatak 3

Implementirati algoritam insertion sort sa potpisom:

```
template <typename It>
void insertionsort(It begin, It end)
```

Zadatak 4

Implementirati algoritam sortiranja merge sort metodom sa potpisom:

```
template <typename It>
void mergesort(It begin, It end)
```

Za razliku od implementacije koja je rađena na vježbama, ova implementacija algoritma treba biti in-place, to jeste ne smije koristiti dodatnu memoriju.

Zadatak 5

Implementirati algoritam sortiranja quick sort metodom sa potpisom:

```
template <typename It>
void quicksort(It begin, It end)
```

Za razliku od implementacije koja je rađena na vježbama, ova implementacija treba da, nakon što rekurzijom smanji obim problema, za male vrijednosti podniza pozove insertionsort.

Zadatak 6

Neka je data struktura Tim definisana na sljedeći način:

```
struct Tim {
  std::string  naziv;
  int bodovi;
  int primljeniGolovi;
  int postignutiGolovi;
};
```

Napisati funkciju koja sortira niz objekata klase Tim proizvoljne dužine tako da budu zadovoljeni sljedeći kriteriji:

- timovi na kraju treba da budu sortirani po broju bodova u rastućem redoslijedu,
- ukoliko postoji više timova sa istim brojem bodova, bolje pozicionirani tim je tim sa boljom gol-razlikom (razlikom postignutih i primljenih golova),
- ukoliko postoji više timova sa istom gol-razlikom, bolje pozicioniran je tim sa većim brojem postignutih golova,
- ukoliko postoji dva ili više timova sa istim brojem postignutih i primljenih golova te istim brojem bodova, u konačnom poretku treba da budu međusobno poredani po rastućem abecednom redoslijedu.

Za riješavanje ovog zadatka potrebno je pozivati sort funkcije sa različitim kriterijima sortiranja. Odabrane sort funkcije implementirati tako da primaju funkciju kao parametar umjesto klasične operacije $<$. Implementirani algoritam treba biti in-place i imati minimalnu moguću složenost. Objasniti implementirano rješenje.

Zadatak 7

Potrebno je implementirati divide and conquer algoritam koji u O(logN) vremenu pronalazi minimalan element u cirkularno pomjerenom sortiranom nizu. Naprimjer, ukoliko imamo sortiran niz A = {1, 2, 3, 4, 5, 6} cirkularno pomjeren niz za 2 elementa je A' = {5, 6, 1, 2, 3, 4}.

Zadatak 8

Implementirati funkciju *pronadjiNajmanji* koja vraća najmanji element proslijeđenog niza korištenjem *divide and conquer* strategije.

```
int pronadjiNajmanji(int*, int);
```

Način predavanja

Zadaću je potrebno predati u obliku arhive *ime_prezime_zadaca1.zip*, pri čemu sadržaj arhive treba da bude direktorij *ime_prezime_zadaca1*, sa poddirektorijem

za svaki zadatak, sa imenima: zadatak1, zadatak2, . . . , zadatak9.

Pored toga, potrebno je da se studenti strogo pridržavaju naziva metoda, njihovih parametara i povratnih vrijednosti u zadacima gdje su oni naglašeni. Za izlazak na provjeru potrebno je implementirati i predati minimalno 50% zadaće.

Sve predane zadaće će biti automatski pregledane za plagijate i svaki pokušaj prepisivanja će biti prijavljen i adekvatno sankcionisan.