Fakultet tehničkih nauka, Novi Sad
Animacija u inženjerstvu

11. II 2019. godine

Diskr. i komb. met. za rač. gr.

1→15, 2→5, 3→10, 4→10, 5→5, 6→15, 7→10, 8→10.

1. Napisati algoritam za sortiranje umetanjem, takozvani INSERTION SORT.

Za algoritam INSERTION SORT iz zadatka 1, za niz dužine n, neka je $S(n)$ broj upisivanja elemenata u niz i $P(n)$ broj poređenja.

2. Za niz $[6, 1, 2, 3, 4, 5]$ naći $S(n)$ i $P(n)$.
3. Koliko je $S(n)$ i $P(n)$ za obrnuto sortirani ulazni niz dužine n algoritma INSERTION SORT?
4. Dati definiciju "velikog O" ponašanja i pokazati da je $\frac{1}{4}\, n \ln n + 40\, n - 10 = O(n\sqrt{n})$.

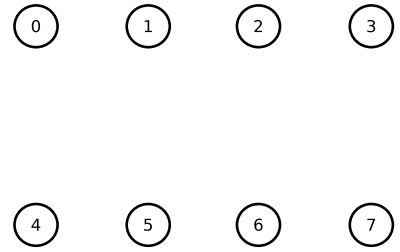
   Za niz dužine n neka je $T_{WM}(n)$ najgori slučaj vremena sortiranja Merge sort algoritmom i $T_{BI}(n)$ najbolji slučaj vremena sortiranja Insertion sort algoritmom. Da li je $T_{WM}(n) = O(T_{BI})$?

   Da li je $\frac{3}{4}\, n^2 + 3\, n\, \sqrt{n} = o(n^2 \ln n)$?

---

5. Nacrtati usmereni graf G koji je dat tabelom listi susedstva:

u	Adj(u)
0	1, 5
1	
2	0, 3, 5, 6
3	7
4	5, 6
5	6
6	7
7	



6. Na graf G primeniti DFS algoritam, kod čvorova napisati d i f vrednosti, kod grana napisati tip (TBCF), napraviti tabelu zagrada. Ako je dati graf DAG, dati topološko sortiranje čvorova, ako nije, dati graf komponenti jake povezanosti.
7. Napisati kod funkcije `adjlist2adjmatrix` koja za graf G sa n čvorova dat nizom povezanih listi susedstava nalazi matricu susedstva M. Dati tabelu listi susedstava i matricu susedstva grafa iz zadatka 5.

---

8. U tabeli su date cene prevoza između 5 gradova.

   (a) Polazeći od čvora 1, metodom najjeftinijeg suseda naći približno rešenje problema trg. putnika (TSP).

   (b) Za isti problem naći Mađarskom metodom angažovanje koje je rešenje relaksiranog TSP. Komentarisati rešenja (a) i (b).

	1	2	3	4	5
1	-	10	6	12	4
2	12	-	7	12	8
3	7	6	-	5	9
4	14	13	8	-	7
5	5	9	9	8	-

```
#define max_cv 50
typedef struct _node gnode;
typedef gnode *grana;
struct _node
{
    int data;
    gnode *next;
};

void adjlist2adjmatrix(grana G[],
  int n, unsigned char M[])
{ /* Ovaj kod napisati */ }
```

```
int main(void)
{   grana G[max_cv];
    int i,n; grana *rear[max_cv];
    unsigned char M[max_cv*max_cv];
    for(i=0;i<max_cv;i++){
        G[i] = NULL;
        rear[i] = &(G[i]);}
    enqueue_list(&rear[0],1);
// ...
    enqueue_list(&rear[6],7); n = 8;
    adjlist2adjmatrix(G, n, M);
    printmatrix(M,n,n);
}
```