Fakultet tehničkih nauka, Novi Sad 9. IX 2020. godine
Animacija u inženjerstvu $10 + 10 + 10 + 10 + 5 + 15 + 10 + 10 = 80$

Diskretne i kombinatorne metode za računarsku grafiku

Dat je algoritam POLINOM za računanje vrednosti $p_n(x) = a_n x^n + a_{n-1}x^{n-1} + \cdots + a_1 x + a_0$, za niz $A = [a_0, a_1, \ldots, a_n]$.

```
1: function POLINOM(A, x)
2:     n ← length(A) − 1
3:     p ← 0
4:     t ← 1
5:     for k ← 0 to n do
6:         p ← p + A[k] · t
7:         t ← t · x
8:     end for
9:     return p
10: end function
```

1. Izračunati broj sabiranja $S(n)$ (linija 6) i broj množenja $M(n)$ (linije 6 i 7) potrebnih da se izračuna vrednost polinoma stepena n.

2. Napisati pseudo kod algoritma HORNER za računanje vrednosti polinoma Hornerovom šemom.

3. Izračunati broj sabiranja $S_H(n)$ i broj množenja $M_H(n)$ potrebnih da se izračuna vrednost polinoma stepena n Hornerovom šemom:

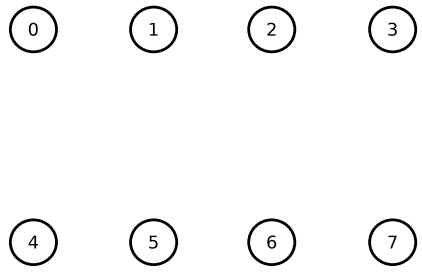
	a_n	a_{n-1}	$\cdots$	a_1	a_0
x	a_n	$x \cdot a_n + a_{n-1}$	$\cdots$	$\cdots$	$p_n(x)$

4. Napisati u programskom jeziku C proceduru za množenje matrica $C_{m\times n} = A_{m\times p} \cdot B_{p\times n}$ koje su smeštene u nizove susednih memorijskih lokacija A, B i C.

```
void multmat(double * A, double * B, double * C, int m, int p, int n);
```

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

u	Adj(u)
0	1
1	2, 4, 5
2	6
3	7
4	0
5	6
6	2, 3, 7
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 enqueue_list koja unosi čvor na kraj liste susedstva grafa. Napisati deo koda za unos grafa G (iz zad. 5) u okviru procedure main u niz listi susedstva grafa G leksikografski.

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.
   (c) Komentarisati rešenja (a) i (b).

	1	2	3	4	5
1	-	7	12	13	8
2	8	-	4	8	9
3	16	5	-	10	6
4	12	9	12	-	7
5	5	9	7	6	-

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

void enqueue_list(grana **gp, cvor d)
{
// Ovaj kod napisati
}
```

```
int main(void)
{   grana G[max_cv],GT[max_cv];
    int i,n; grana *rear[max_cv];

    for(i=0;i<max_cv;i++){
        G[i] = NULL;
        rear[i] = &(G[i]);
    }
    enqueue_list(&rear[0],1);
    // nastaviti dalje,
    // uneti ostatak grafa
    return 0;
}
```