

BT1. Xác định độ phức tạp tính toán của các đoạn code dưới đây

a.

```
int sum = 0;
for (int i = 0; i < n; i++)
    for (int j = 0; j < n * n; j++)
        sum++;
```

-> $O(n^3)$

b.

```
int sum = 0;
for (int i = 0; i < n; i++)
    for (int j = 0; j < i; j++)
        sum++;
```

-> $O(n^2)$

c.

```
int sum = 0;
for (int i = 0; i < n; i++)
    for (int j = 0; j < i*i; j++)
        for (int k=0; k<j; k++)
            sum++;
```

-> $O(n^5)$

d.

```
int sum = 0;
for (int i = 0; i < n; i++)
    sum++;
int val = 1;
for (int j = 0; j < n * n; j++)
    val = val * j;
```

-> $O(n^2)$

e.

```
int sum = 0;
for (int i = 0; i < n; i++)
```

```

    sum++;
for (int j = 0; j < n * n; j++)
    computer_val (sum, j);

```

Biết thời gian tính toán của hàm computer_val (x, y) là $O(n \log n)$

-> $O(n^3 \log n)$

f.

```

int s = 0;
for (int i = 1; i < n - 2; i++)
    for (int j = i + 1; j < n - 1; j++)
        for (int k = j + 1; k <= n; k++)
            s = s + 1;

```

-> $O(n^3)$

g.

```

int s = 0;
for (int i = 1; i < n - 2; i++)      ->n
    for (int j = n; j > 0; j=j/2)    ->logn
        for (int k = j + 1; k <= n; k++) ->n
            s = s + 1;

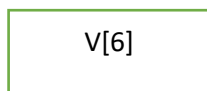
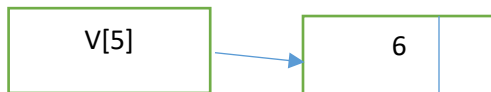
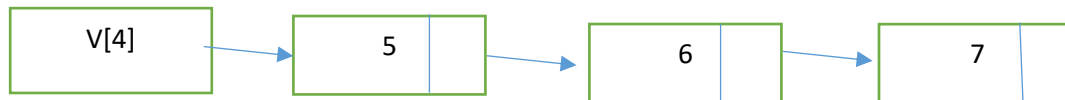
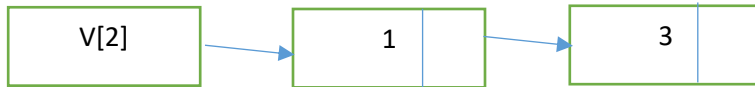
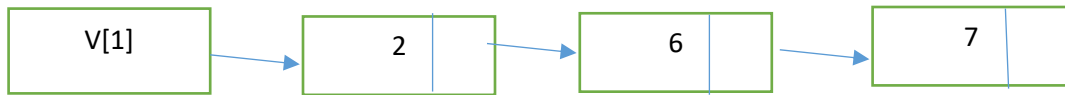
```

-> $O(n^2 \log n)$

BT2. Xét đồ thị có hướng G được đánh số từ 1 đến 7 được cho bởi ma trận kề A sau đây:

0	1	0	0	0	1	1
1	0	1	0	0	0	0
1	0	0	1	0	0	0
0	0	0	0	1	1	1
0	0	0	0	0	1	0
0	0	0	0	0	0	0
0	0	1	0	0	1	0

a. Đưa ra danh sách kề của đồ thị đã cho



b. Thực hiện tìm kiếm theo chiều sâu trên đồ thị G (giả thiết là khi duyệt các đỉnh của đồ thị và các đỉnh trong danh sách kề của một đỉnh, ta duyệt theo thứ tự chỉ số tăng dần).

Lưu ý: Làm bài tập 2 mà không vẽ đồ thị.

$\text{DFS}(G, 1) = 1, 2, 3, 4, 5, 6, 7$

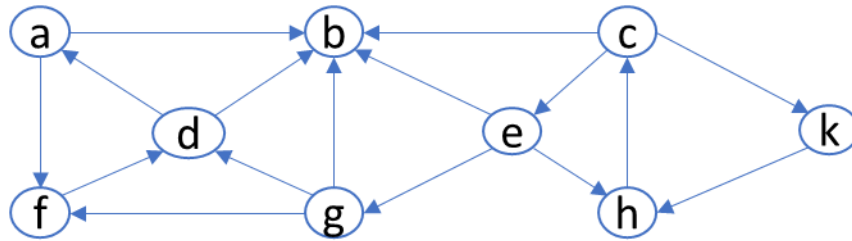
$\text{DFS}(G, 2) = 2, 1, 6, 7, 3, 4, 5$

$\text{DFS}(G, 3) = 3, 1, 2, 6, 7, 4, 5$

$\text{DFS}(G, 4) = 4, 5, 6, 7, 3, 1, 2$

$\text{DFS}(G, 7) = 3, 1, 2, 6, 7, 4, 5$

BT3. Cho đồ thị có hướng G như hình vẽ



a. Đưa ra ma trận kề của đồ thị

0	1	0	0	0	1	0	0	0
0	0	0	0	0	0	0	0	0
0	1	0	0	1	0	0	0	1
1	1	0	0	0	0	0	0	0
0	1	0	0	0	0	1	1	0
0	0	0	1	0	0	0	0	0
0	1	0	1	0	1	0	0	0
0	0	1	0	0	0	0	0	0
0	0	0	0	0	0	0	1	0

b. Thực hiện thuật toán tìm kiếm theo chiều sâu trên G (giả thiết là khi duyệt đỉnh, ta duyệt đỉnh theo thứ tự trong bảng chữ cái)

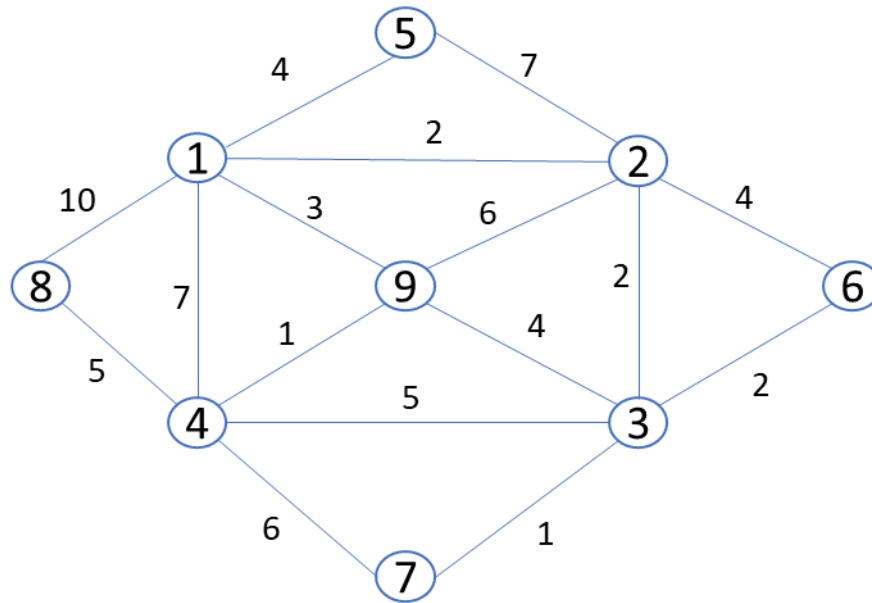
$\text{DFS}(G, c) = c, b, e, g, d, a, f, h, k$

$\text{DFS}(G, e) = e, b, g, d, a, f, h, c, k$

$\text{DFS}(G, h) = h, c, b, e, g, d, a, f, k$

$\text{DFS}(G, k) = k, h, c, b, e, g, d, a, f$

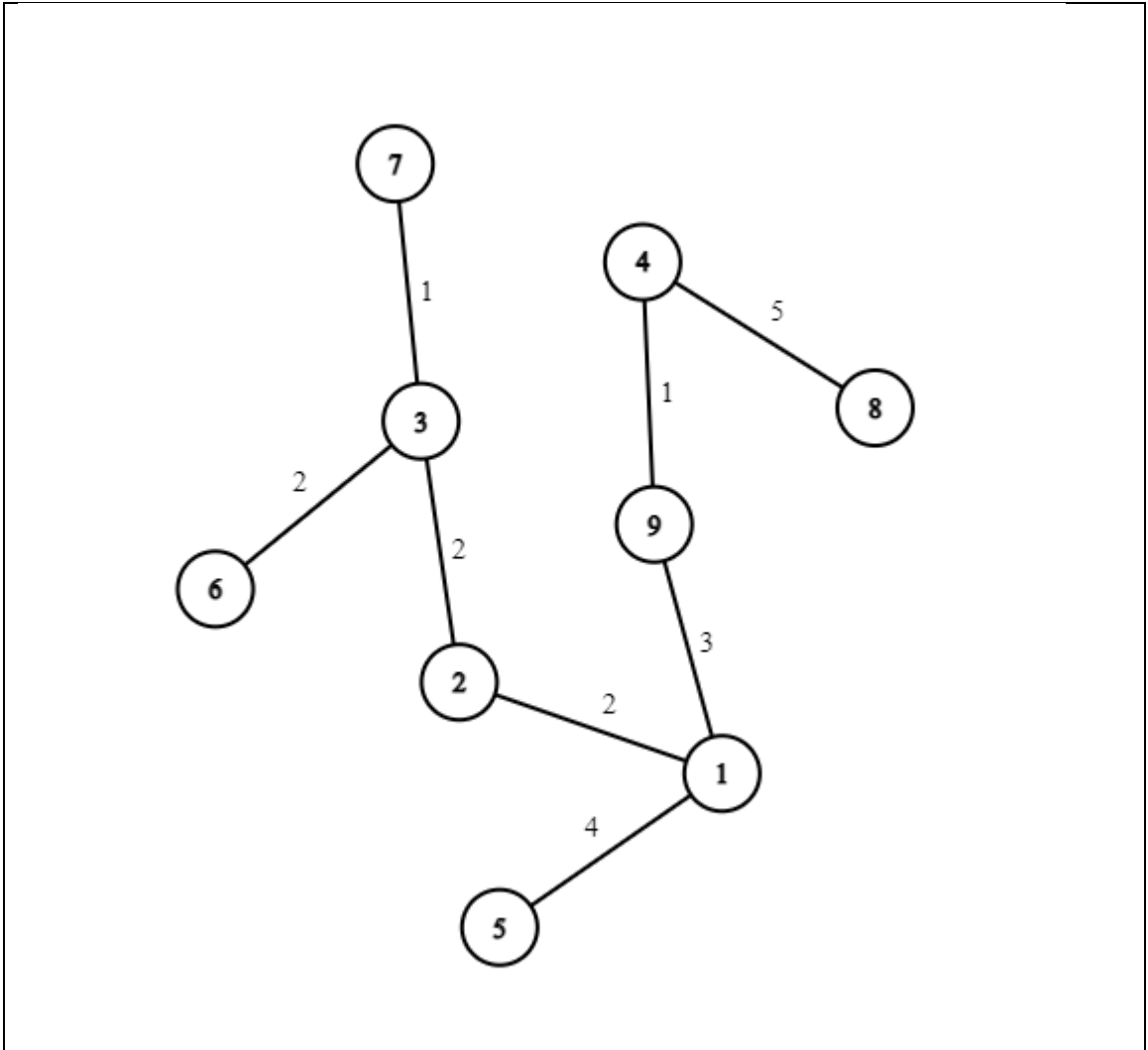
BT4. Cho đồ thị dưới đây



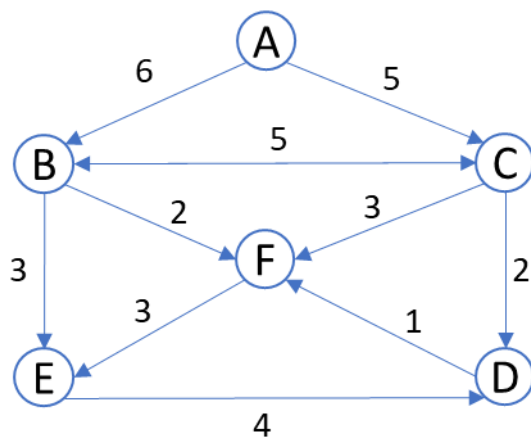
a. Hãy đưa ra ma trận trọng số của đồ thị

0	2	0	7	4	0	0	10	3
2	0	2	0	7	4	0	0	6
0	2	0	5	0	2	1	0	4
7	0	5	0	0	0	6	5	1
4	7	0	0	0	0	0	0	0
0	4	2	0	0	0	0	0	0
0	0	1	6	0	0	0	0	0
10	0	0	5	0	0	0	0	0
3	6	4	1	0	0	0	0	0

b. Tìm cây khung nhỏ nhất của đồ thị bằng thuật toán Kruskal



BT5. Cho đồ thị dưới đây



Tìm đường đi ngắn nhất từ đỉnh A đến tất cả các đỉnh còn lại trên đồ thị theo thuật toán Dijkstra.

Bảng bước lặp	A			B			C			D			E			F		
	d	p	k	d	p	k	d	p	k	d	p	k	d	p	k	d	p	k
Khởi tạo	∞	-	F	∞	-	F	∞	-	F	∞	-	F	∞	-	F	∞	-	F
1	0	-	T	6	A	F	5	A	F	∞	-	F	∞	-	F	∞	-	F
2				6	A	F	5	A	T	7	C	F	∞	-	F	8	C	F
3				6	A	T				7	C	F	9	B	F	8	C	F
4										7	C	T	9	B	F	8	C	F
5													9	B	F	8	C	T
6													9	B	T			