VNUHCM - University of Science

fit@hcmus

CSC10004 – Data Structures and Algorithms

Session 10
Review

Instructors:
Dr. Lê Thanh Tùng

Efficiency of Algorithm

- What does the function `F1()` do?

```
int F1(int n){
    int S = 0;
    for(int i = 1; i <= n; i++){
        S = S + i;
    }
    return S;
}
```

- Analyze the time complexity (in Big-O notation) of `F1()`

```
int F1(int n){
    int S = 0;
    for(int i = 1; i <= n; i++){
        S = S + i;
    }
    return S;
}
```

- What does the function `F1()` do? → The function F1() calculates the sum of integers from 1 to n
- Analyze the time complexity (in Big-O notation) of `F1()`
  - Choose assignment S = S + i as a basic operation
  - Number of assignment operations: $\sum_{i=1}^{n} 1 = n - 1 + 1 = n$
    - So, the time complexity belongs to $\boldsymbol{O(n)}$

- Consider the array A = {29, 18, 10, 15, 20, 9, 5, 13, 2, 4, 15}.

  a. Does A satisfy the max-heap property? If not, fix it by swapping two elements

  b. Utilize the Heap Sort algorithm until finding the third max-element

  c. What is the value of the pivot element when using the median-of-three pivot selection technique in the given array?

  d. Assume that 13 is chosen as the pivot of the given array. What is the array after the first partition?

- Consider the array A = {29, 18, 10, 15, 20, 9, 5, 13, 2, 4, 15}.

  a. Does A satisfy the max-heap property? If not, fix it by swapping two elements

  - A doesn’t satisfy the max-heap because a[1] < a[2*1 + 2]
  - Swapping two numbers 18 and 20 will fix the array A to max-heap
    - After modifying it: A = {29, 20, 10, 15, 18, 9, 5, 13, 2, 4, 15}

- Consider the array A = {29, 18, 10, 15, 20, 9, 5, 13, 2, 4, 15}.

  b. Utilize the Heap Sort algorithm until finding the third max-element

  - Max-heap: {29, 20, 10, 15, 18, 9, 5, 13, 2, 4, 15}
  - Swap(29, 15): {15, 20, 10, 15, 18, 9, 5, 13, 2, 4} | {29}
  - Heapify a[0]: {20, 18, 10, 15, 15, 9, 5, 13, 2, 4} | {29}
  - Swap(20, 4): {4, 18, 10, 15, 15, 9, 5, 13, 2} | {20, 29}
  - Heapify a[0]: {18, 15, 10, 13, 15, 9, 5, 4, 2} | {20, 29}
  - Swap(18, 2): {2, 15, 10, 13, 15, 9, 5, 4, 2} | {18, 20, 29}
  - The third max-element is: 18

- Consider the array A = {29, 18, 10, 15, 20, 9, 5, 13, 2, 4, 15}.

  c. What is the value of the pivot element when using the median-of-three pivot selection technique in the given array?

  - As utilizing median of three pivot, the pivot is the median of (a[0] = 29, a[11/2] = a[5] = 9, a[10] = 15). So, the pivot is 15

- Consider the array A = {29, 18, 10, 15, 20, 9, 5, 13, 2, 4, 15}.

  d. Assume that 13 is chosen as the pivot of the given array. What is the array after the first partition?

  - 13, 18, 10, 15, 20, 9, 5, 29, 2, 4, 15 (a0 -> a7)
  - 13 | 10 | 18, 15, 20| 9, 5, 29, 2, 4, 15 (a1 -> a0)
  - 13 | 10, 9| 15, 20, 18| 5, 29, 2, 4, 15 (a5 -> a2)
  - 13 | 10, 9, 5 | 20, 18, 15 29 | 2, 4, 15 (a6 -> a3)
  - 13 | 10, 9, 5, 2 | 18, 15, 29, 20 | 4, 15 (a8 -> a4)
  - 13 | 10, 9, 5, 2, 4 | 15, 29, 20, 18, 15 | (a9 -> a5)
  - 4, 10, 9, 5, 2, | 13 |, 15, 29, 20, 18, 15 (a0 -> a5)

- A new sorting algorithm, called "X-sort", is used to sort pairs of positive integers by the magnitudes of the products they form. Initially, the array of pairs has the following order:

  (2,6), (1,2), (2,10), (2,2), (3,4), (5,4)

- After sorting with "X-sort", the array of pairs has this order:

  (1,2), (2,2), (3,4), (2,6), (5,4), (2,10)

- Is "X-sort" a stable sort? Explain your answer

- A new sorting algorithm, called "X-sort", is used to sort pairs of positive integers by the magnitudes of the products they form. Initially, the array of pairs has the following order:

  (2,6), (1,2), (2,10), (2,2), (3,4), (5,4)

- After sorting with "X-sort", the array of pairs has this order:

  (1,2), (2,2), (3,4), (2,6), (5,4), (2,10)

- Is "X-sort" a stable sort? Explain your answer
  - “X-sort” isn’t a stable sort
  - Because the “X-sort” doesn’t preserve the relative order of elements equal products. Two elements (2, 6) and (3, 4) are equal because they have the same products (12). Before sorting, the element (2, 6) stood in front of the element (3, 4). But after sorting, the element (2, 6) stood behind the element (3, 4).

a. Insert the following characters

I, N, F, O, R, M, A, T, I, O, N, T, E, C, H, N, O, L, O, G, Y

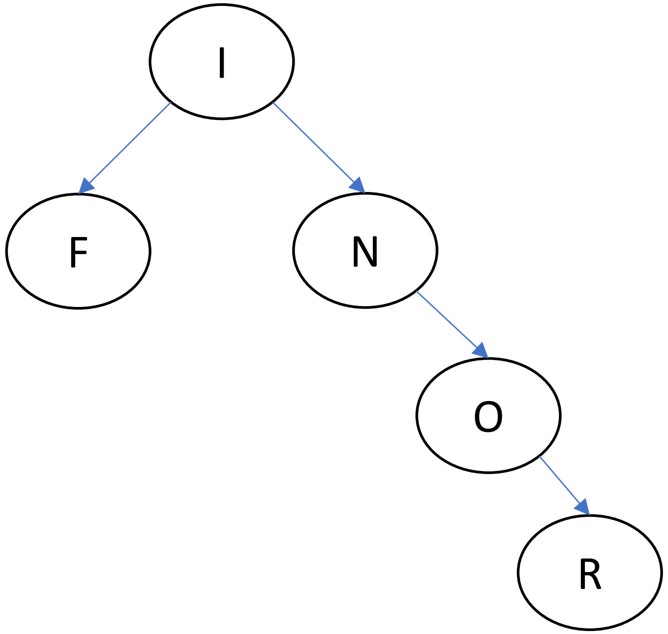
into an empty AVL tree, respectively

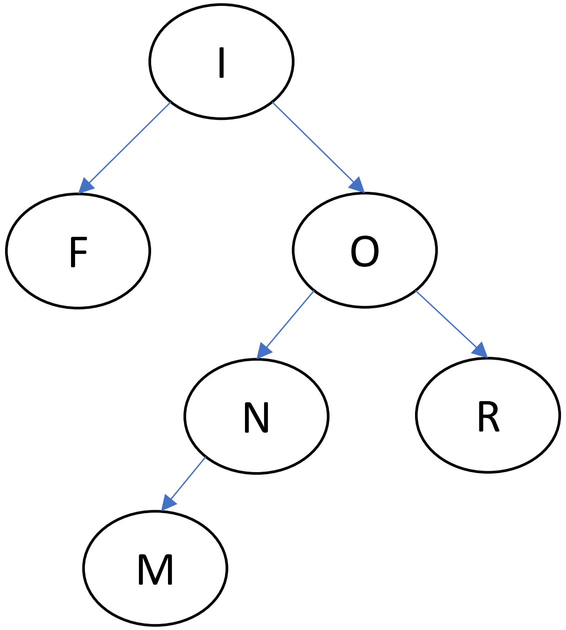
A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z.

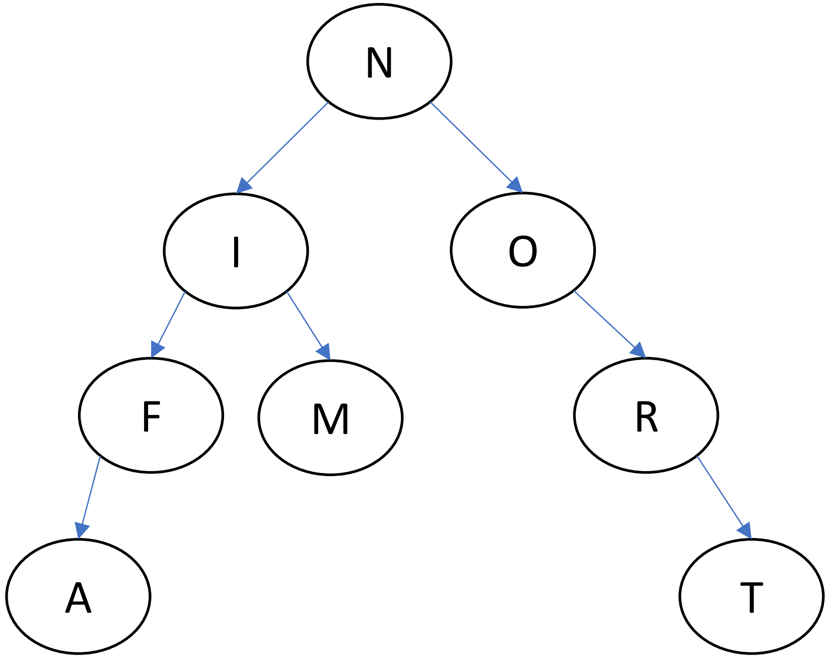
b. Remove the characters E, C, I from the AVL tree from (a), respectively

a. Insert the following characters

I, N, F, O, R, M, A, T, I, O, N, T, E, C, H, N, O, L, O, G, Y

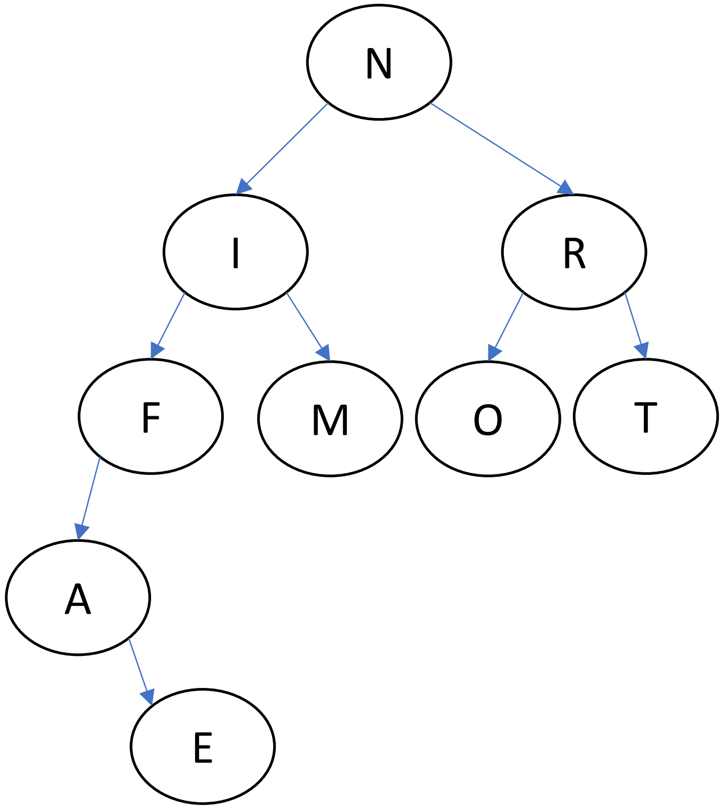


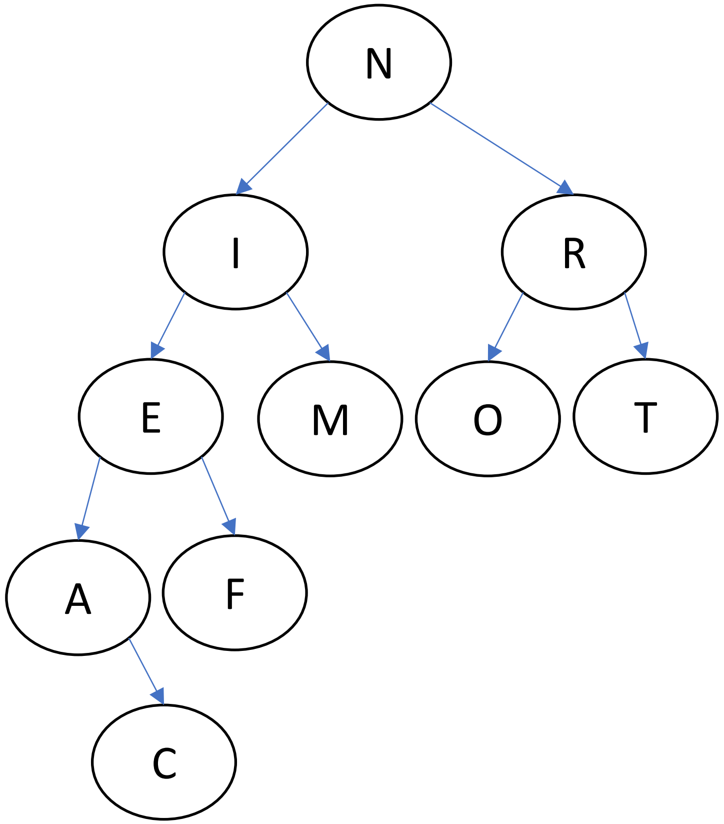


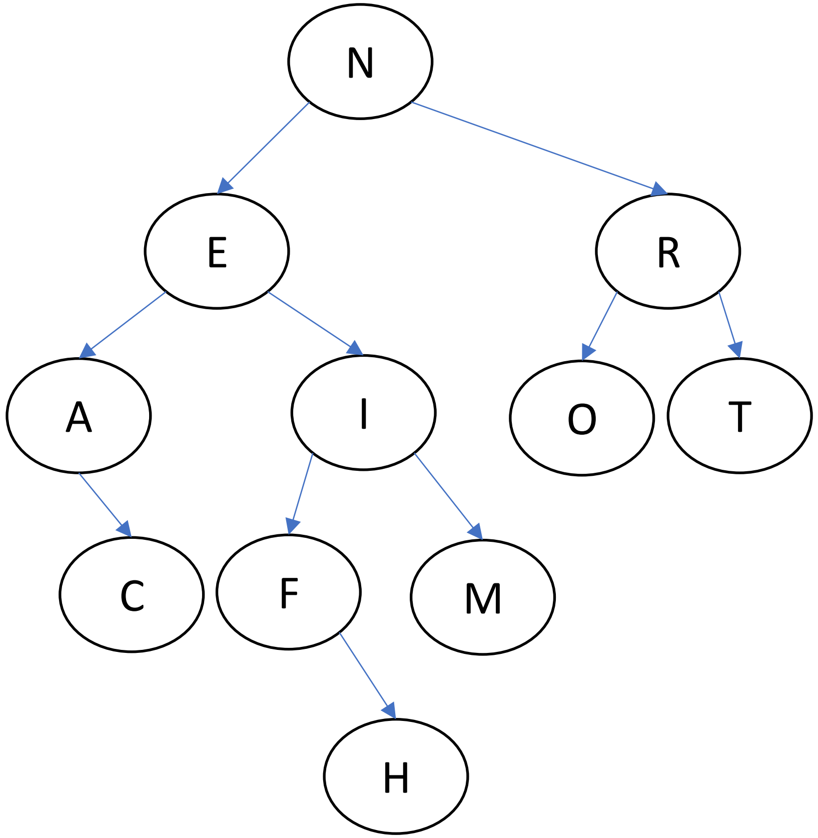


a. Insert the following characters

I, N, F, O, R, M, A, T, I, O, N, T, E, C, H, N, O, L, O, G, Y

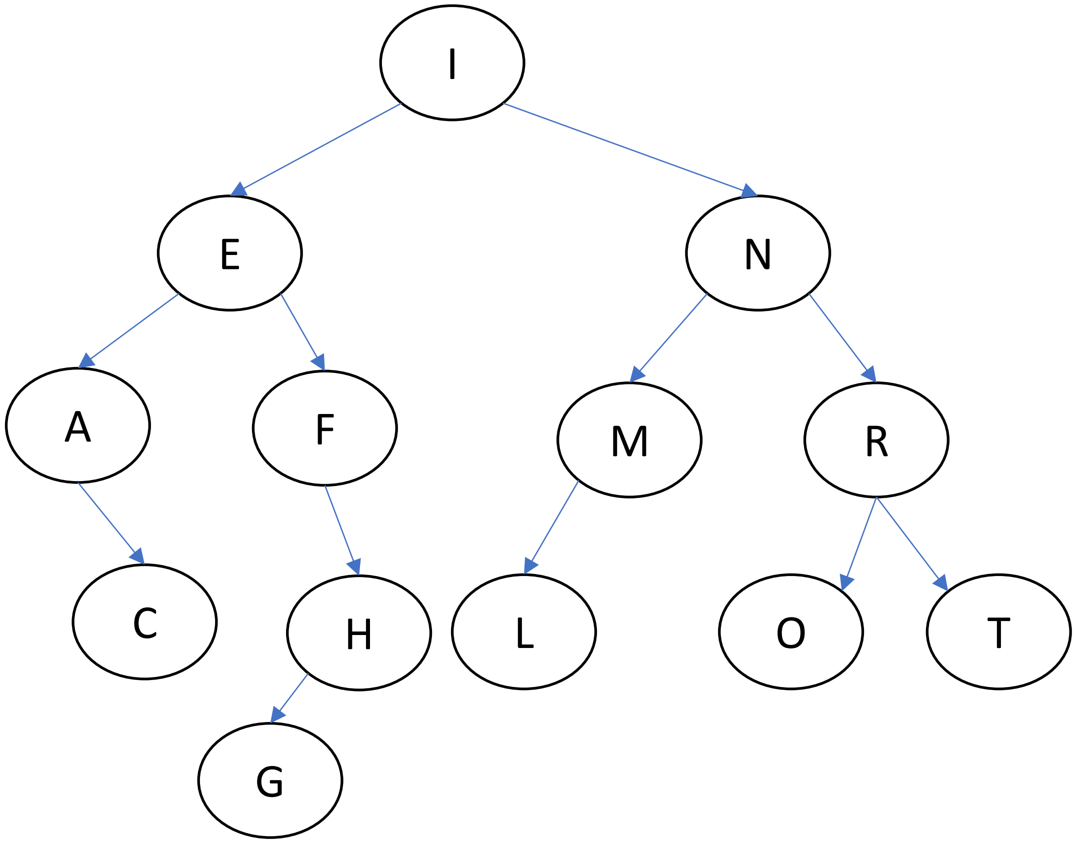


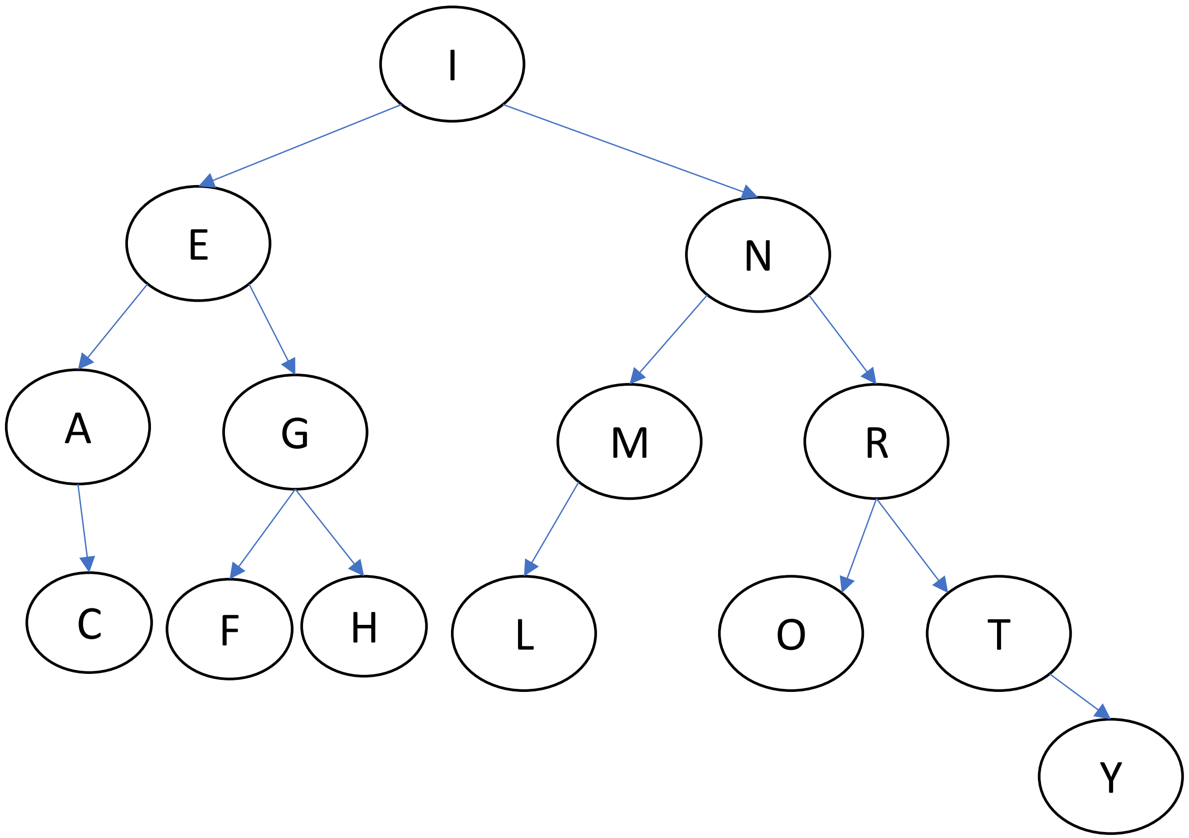




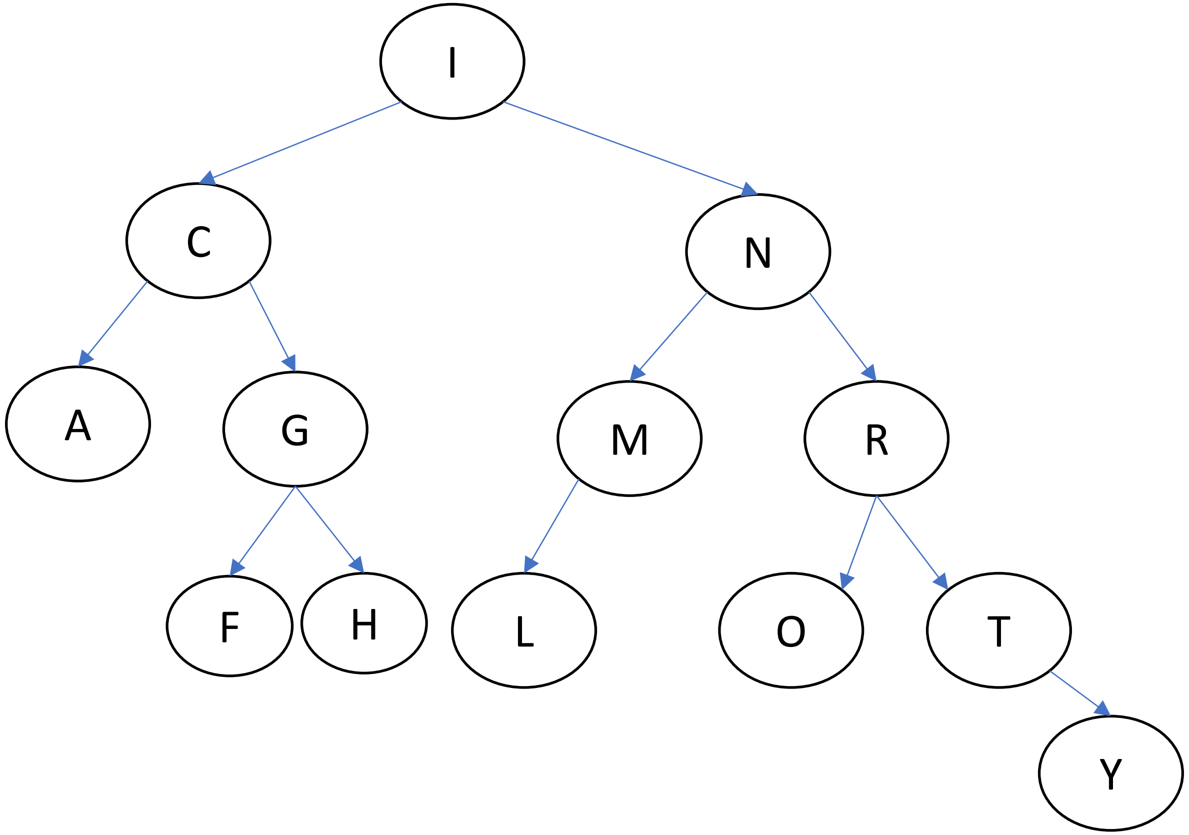
a. Insert the following characters

I, N, F, O, R, M, A, T, I, O, N, T, E, C, H, N, O, L, O, G, Y

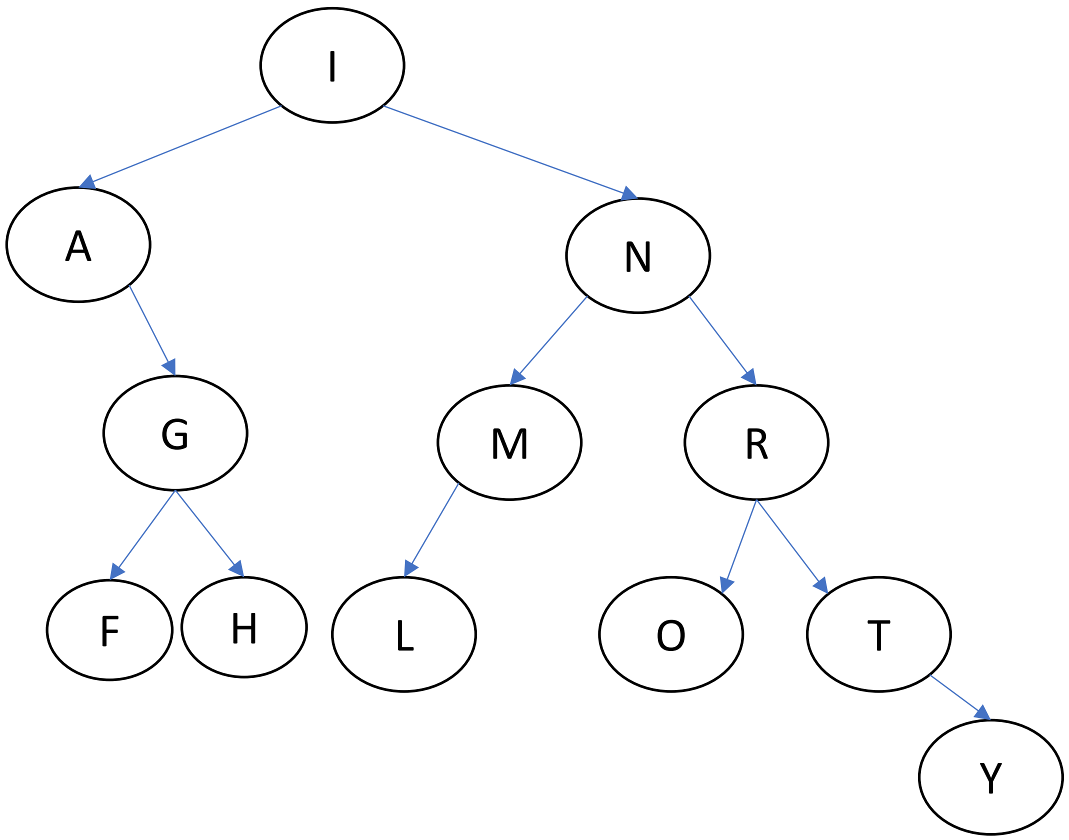


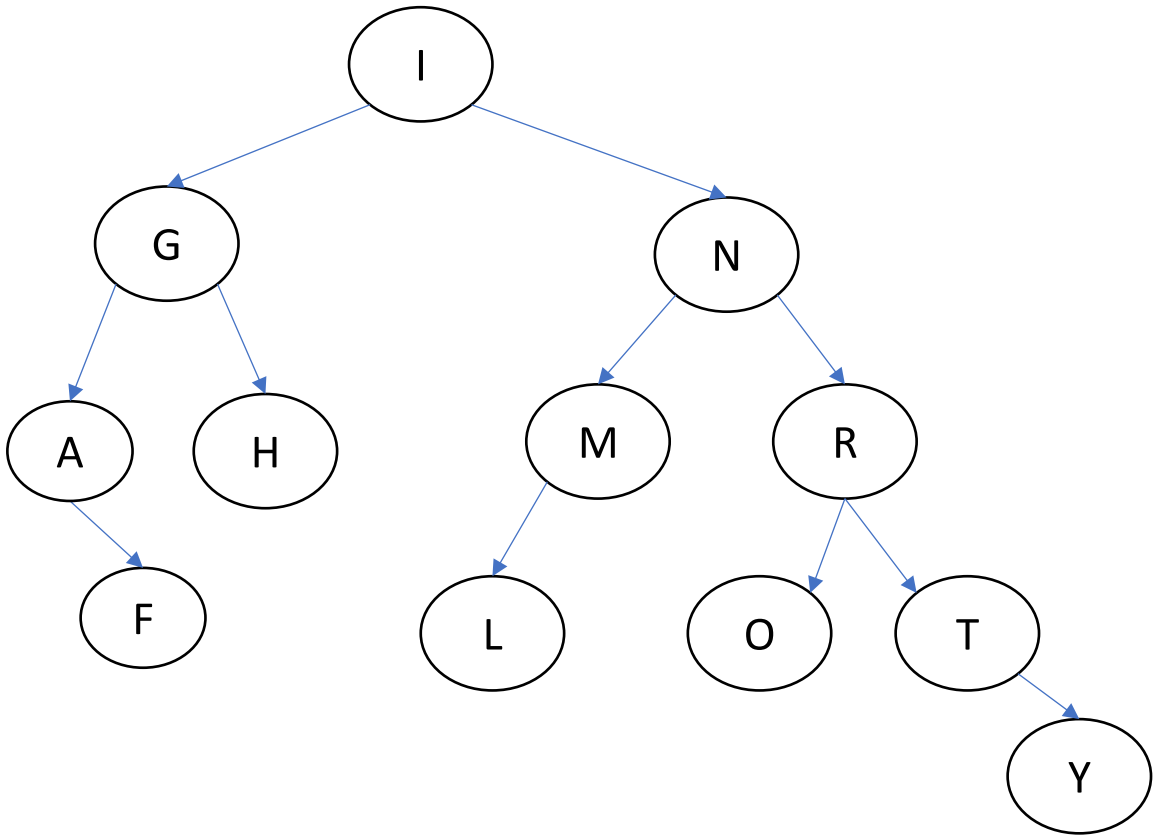


Remove E

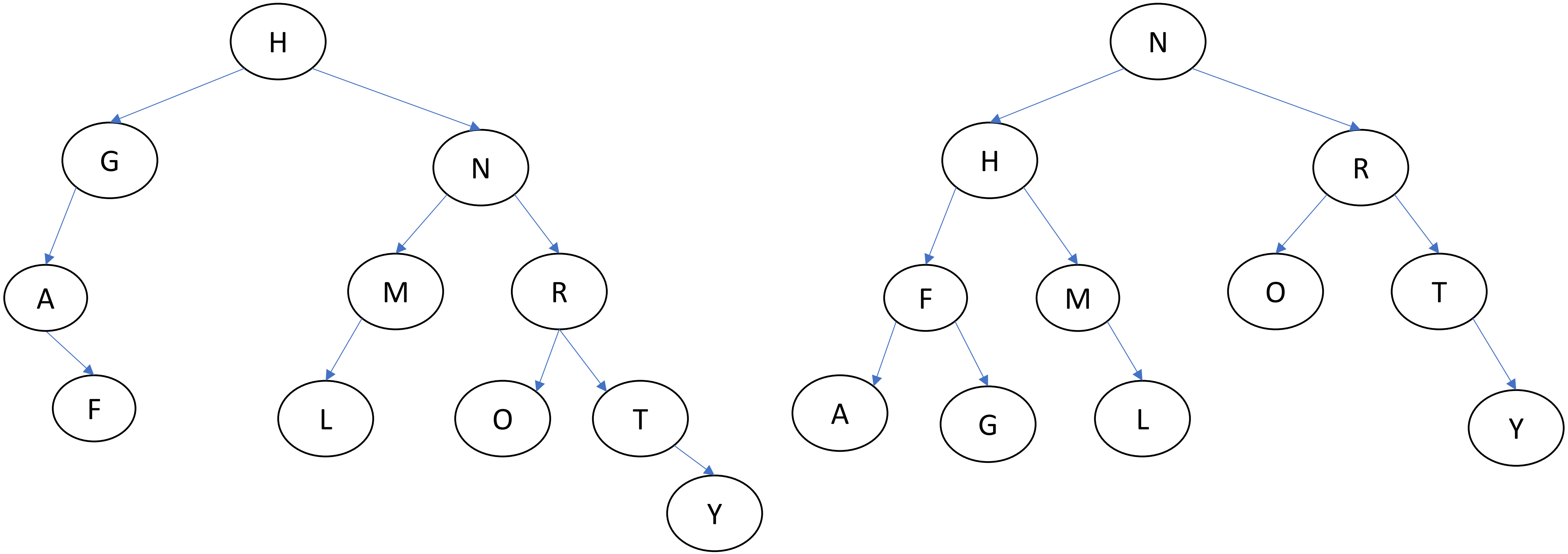


Remove C





Remove I

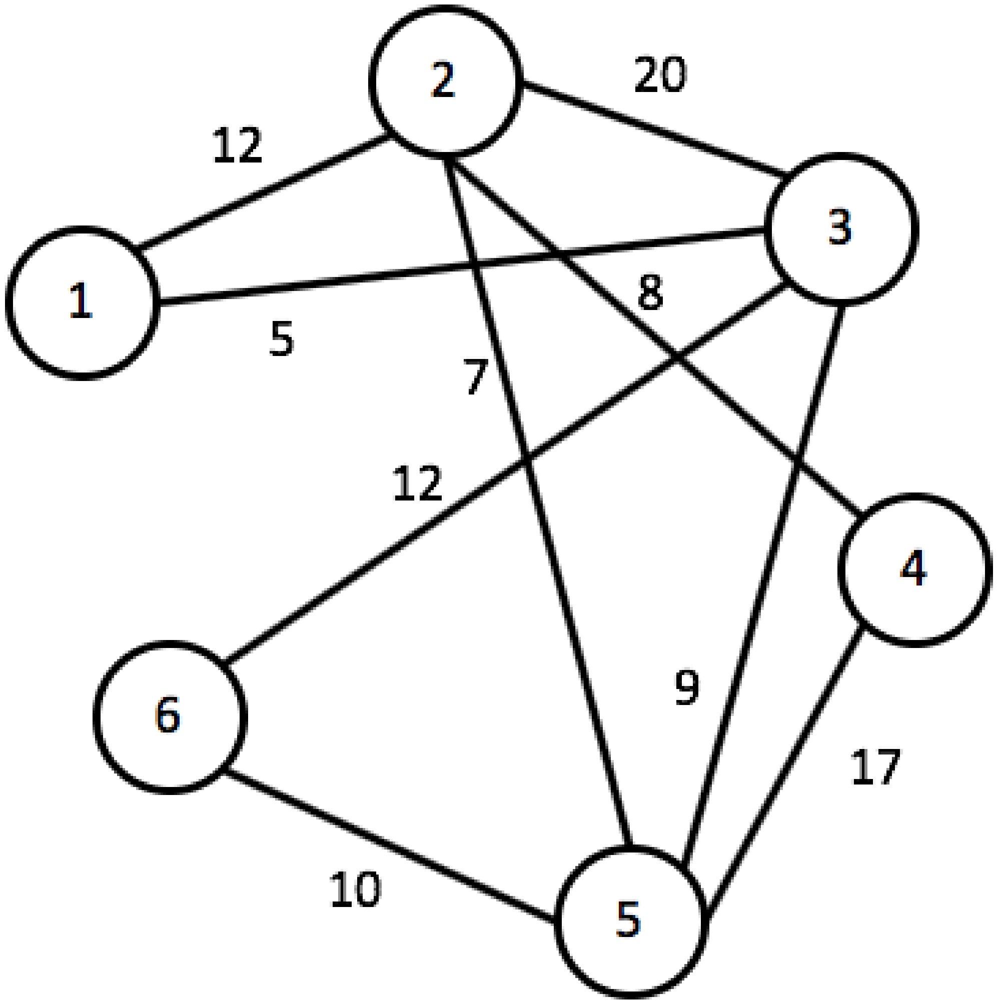


a. Build the adjacency matrix for the graph G

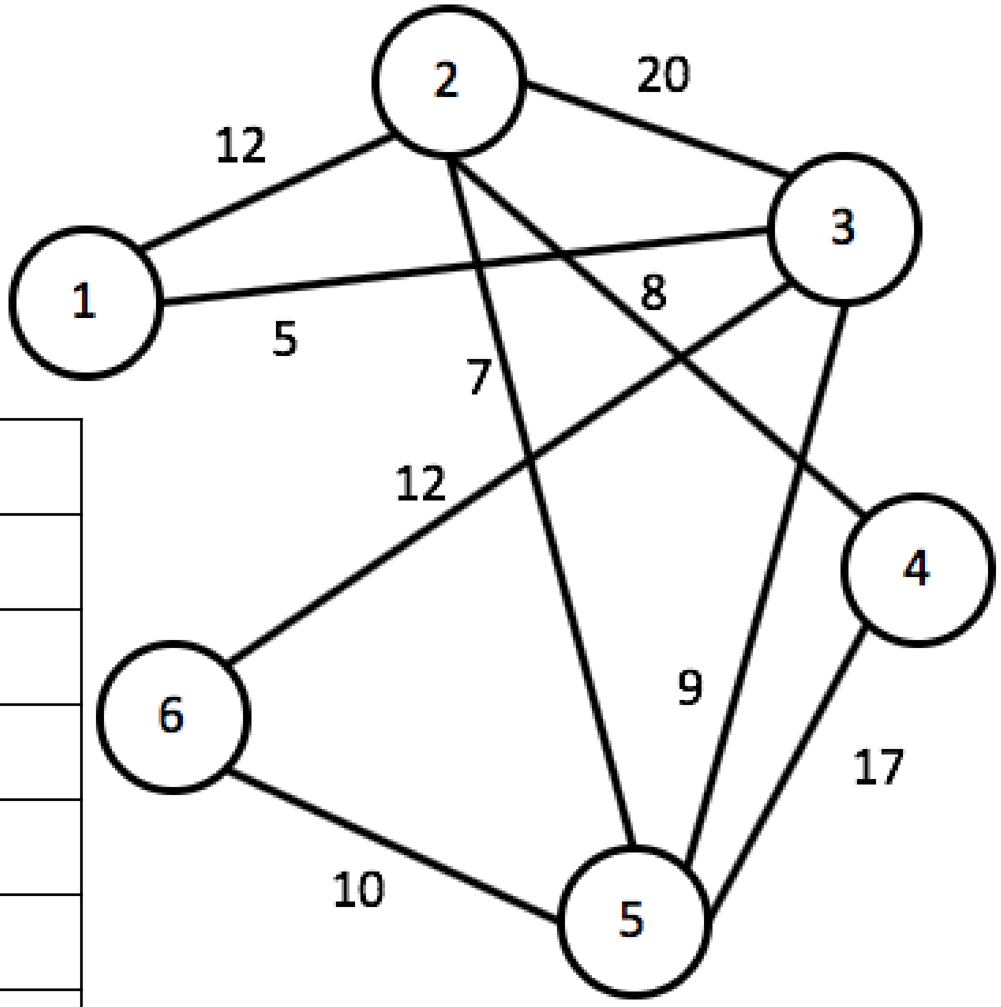
b. Starting from vertex [2] (in the graph G), write down the sequence of visited vertices using the BFS and DFS traversal

c. Starting from vertex [3], illustrate the steps to find the shortest path to other vertices by Dijkstra's Algorithm

d. Starting from vertex [2], find the minimum spanning tree of G by Prim's Algorithm

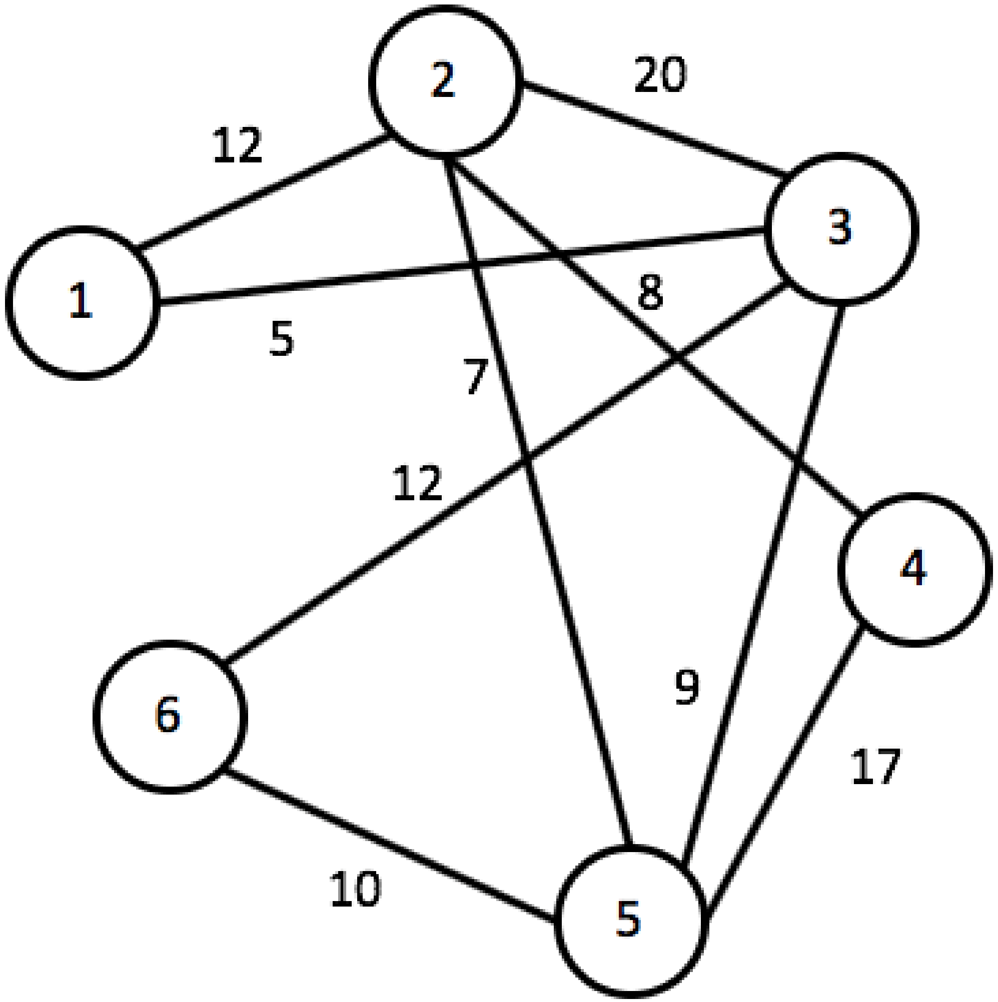


a. Build the adjacency matrix for the graph G



Vertex	1	2	3	4	5	6
1	∞	12	5	∞	∞	∞
2	12	∞	20	8	7	∞
3	5	20	∞	∞	9	12
4	∞	8	∞	∞	17	∞
5	∞	7	9	17	∞	10
6	∞	∞	12	∞	10	∞

b. Starting from vertex [2] (in the graph G), write down the sequence of visited vertices using the BFS and DFS traversal

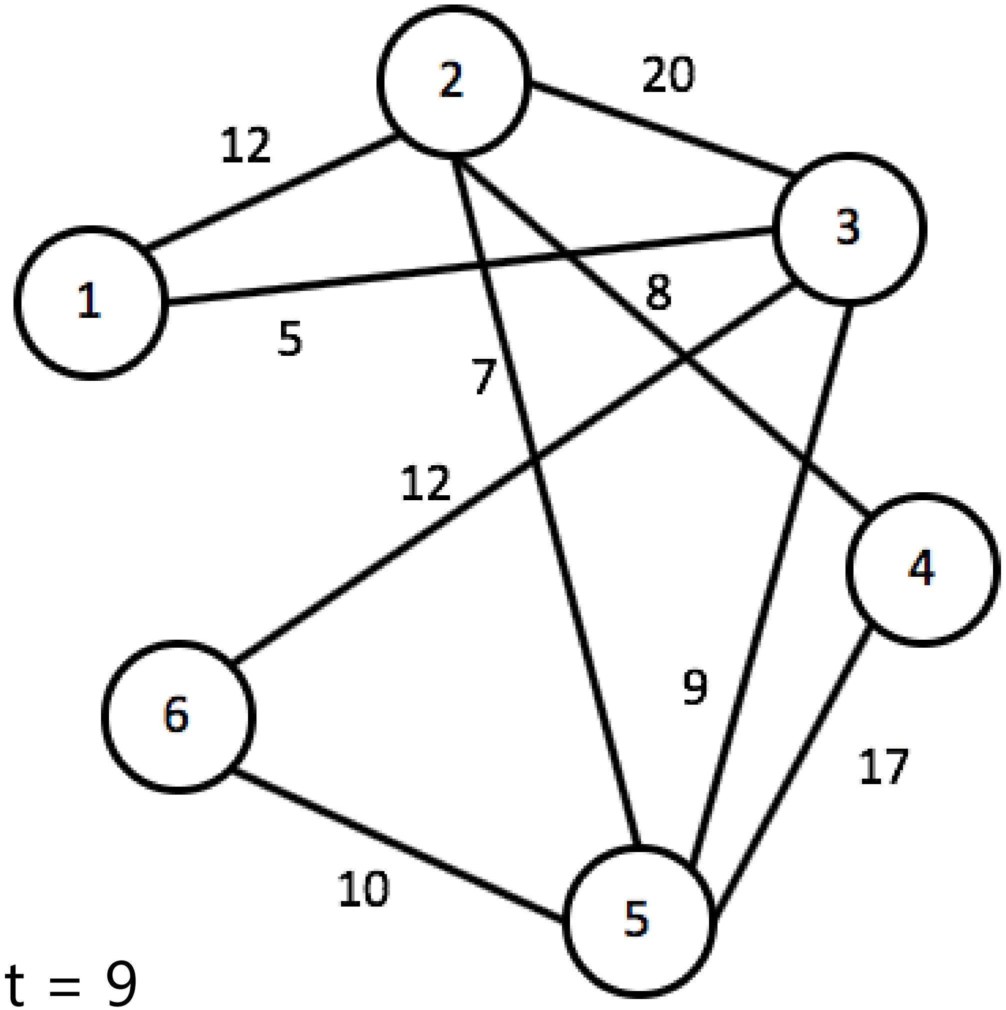


BFS: 2, 1, 3, 4, 5, 6

DFS: 2, 1, 3, 5, 4, 6

c. Starting from vertex [3], illustrate the steps to find the shortest path to other vertices by Dijkstra's Algorithm

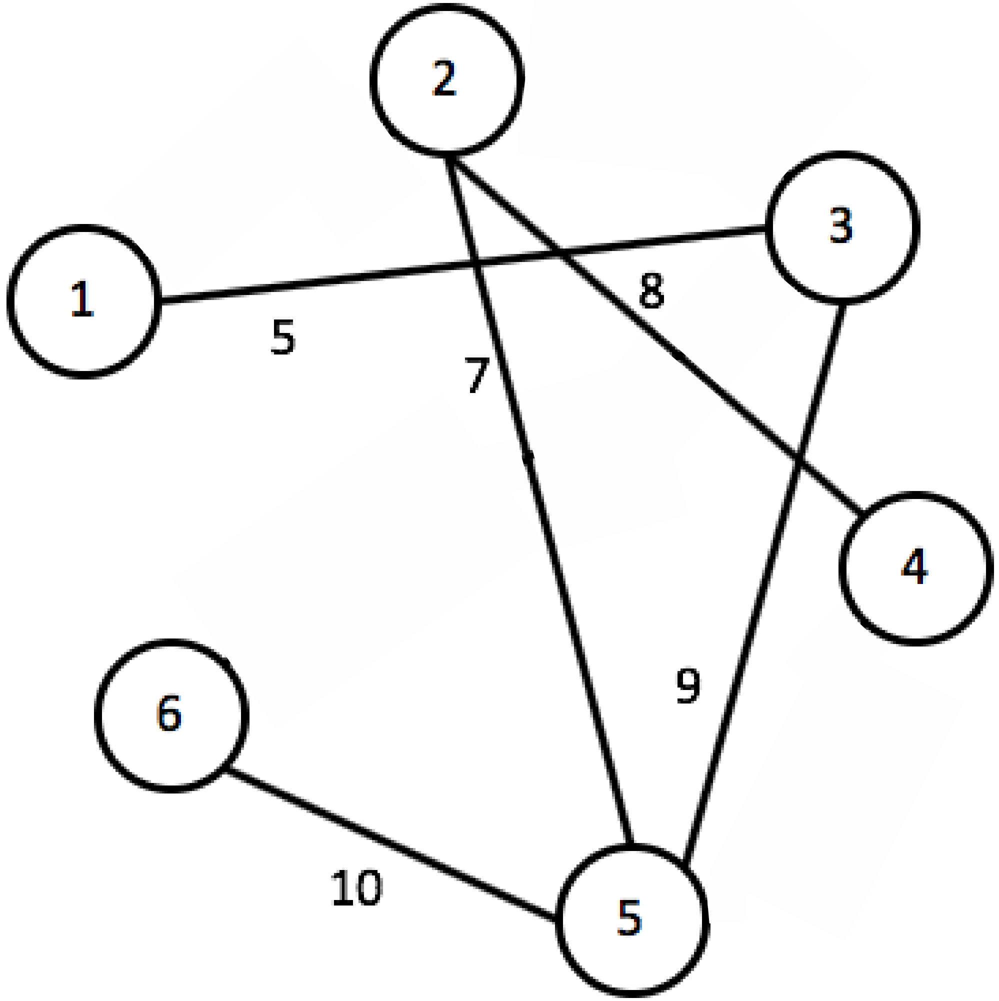
1	2	3	4	5	6
∞, –	∞, –	0	∞, –	∞, –	∞, –
5, 3	20, 3	_	∞, –	9, 3	12, 3
_	17, 1		∞, –	**9, 3**	12, 3
	16, 5		26, 5	_	**12, 3**
	16, 5		26, 5		_
	_		24, 2		



- 3 → 1: Weight = 5
- 3 → 5 → 2: Weight = 16
- 3 → 5 → 2 → 4: Weight = 24
- 3 → 5: Weight = 9
- 3 → 6 Weight = 12

d. Starting from vertex [2], find the minimum spanning tree of G by Prim's Algorithm

Step	Edges	Weight
1	2, 5	7
2	2, 4	8
3	5, 3	9
4	3, 1	5
5	5, 6	10
Total Weight		39



Given an empty hash table with length m = 17.

Hash function: `h(k) = k mod m`.

Show the result after adding these elements:

10, 22, 31, 4, 47, 44, 39, 65, 73

into the hash table, using these given collision-handling methods, respectively

a. Linear probing

b. Quadratic probing: $h(k, i) = h(k) + i^2$

c. Double hashing: $h_2(k) = 1 + (k \bmod 5)$

Hash Table

	Linear Probing	Quadratic Probing	Double Hashing
0			
1			
2			
3			39
4	4	4	4
5	22	22	22
6	39	39	
7	73		
8			
9		73	73
10	10	10	10
11	44	44	
12			
13	47	47	47
14	31	31	31
15	65	65	44
16			65

THANK YOU

for YOUR ATTENTION