

DSP Lab

Graph Algorithms

Satyam Kumar
ID- 20MCMB22

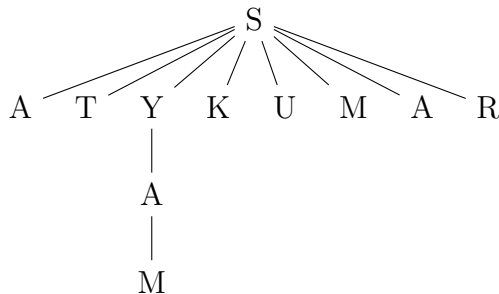
1. Write Algorithms for DFS, BFS, Warshall and Floyd-warshall finding shortest paths. Mention your name and roll number in each of the algorithms.

Algorithm 1 General Skelton code using Input: Name or ID for all algo

► Just replace function call with required function

- 1: **if** Input is String **then** DFS(G,S) ► For name, No input mapping required
- 2: **else** ► For ID(which is alphanumeric), input mapping is required
- 3: Map Letters into integer as A=00,B=01,C=02,D=03,...,Z=25
- 4:
- 5: **if** Digit is found **then** Push that digit into stack
- 6: DFS(G,2)
- 7: Convert the mapped output into alphanumeric character

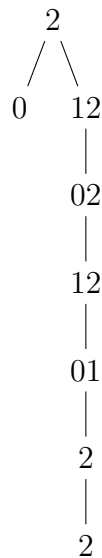
If I provide my name Satyam Kumar as input, Input:SATYAMKUMAR,
Output: ATMAYSKUMAR



Algorithm 2 DFS code for Name or ID

```
1: function DFS( $G, v$ )      ▶  $G$  is the input graph,  $v$  is starting vertex
2:   Initialize stack to be empty ie  $S = \emptyset$ 
3:   Initialize visited array to be  $0$ 
4:   Push the first root into stack
5:   Mark the first node as visited.
6:   while Stack is not empty do
7:      $v \leftarrow$  Top of stack
8:     Pop the top of stack      ▶ Pop a vertex from stack to visit next
9:     while All neighbours  $w$  of  $v$  in Graph  $G$  is not explored do
10:      if  $w$  is not visited then
11:        Push that vertex  $w$  into stack
12:        mark  $w$  as visited
```

If I provide my ID 20MCMB22 as input to DFS.
Mapping M=12, C=02,B=01. Stack = 2,0,12,02,12,2,2
DFS tree for the input looks the following:

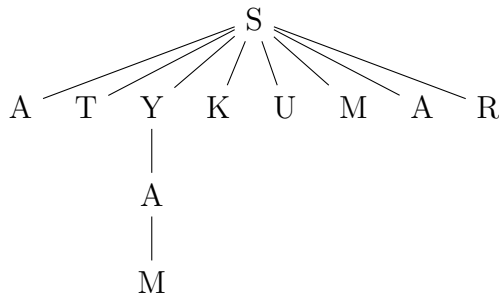


Output (Using DFS):0,2,2,01,12,02,12,2
Output(Alphanumeric): 022MBCM2

If I provide my name without space Satyam Kumar as input to BFS, Input:SATYAMKUMAR, Output: SATYKUMARAM

Algorithm 3 BFS code for Name or ID

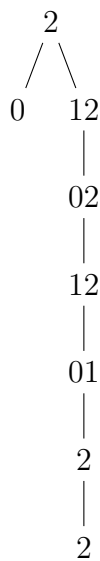
```
1: function BFS( $G, v$ )           ▶  $G$  is the input graph,  $v$  is starting vertex
2:   Initialize queue to be empty ie  $Q = \phi$ 
3:   Enqueue the first root into queue  $Q$  ie  $Q.enqueue(v)$ 
4:   Mark the first node (or root) as visited.
5:   while Queue is not empty do
6:      $v \leftarrow$  Dequeue of stack ▶ Removing that vertex from queue, whose
       neighbour will be visited now
7:     for All neighbours  $w$  of  $v$  in Graph  $G$  is not explored do
8:       if  $w$  is not visited then
9:         Enqueue that vertex  $w$  into stack
10:        mark  $w$  as visited
```



If I provide my ID 20MCMB22 as input.

Mapping M=12, C=02, B=01. Queue = 2,0,12,02,12,2,2

BFS tree for the input looks the following:



Output (Using BFS):2,0,12,02,12,01,2,2

Output(Alphanumeric): 20MCMB22

Algorithm 4 Warshall Pseudocode for Name or ID

Input Adjacency matrix of mapped (character,index)

Output Transitive closure of adjacency matrix of Name or ID

Example Input:SATYAM

Mapped Input: $(S, 0), (A, 1), (T, 2), (Y, 3), (A, 4), (M, 5)$

```
1: function WARSHALL( $adj[1..V][1..V]$ )  $\triangleright$   $adj$  is the adjacency matrix
   of  $V \times V$  dimension
2:    $R^0 \leftarrow adj$   $\triangleright R^0$  is adjacency matrix of initial graph
3:   for  $k \leftarrow 1$  to  $V$  do
4:     for  $i \leftarrow 1$  to  $V$  do
5:       for  $j \leftarrow 1$  to  $V$  do
6:          $R^k(i, j) \leftarrow R^{k-1}(i, j)$  or  $R^{k-1}(i, k)$  and  $R^{k-1}(k, j)$ 
```

If Input is provided as ID 20MCMB22, which is mapped as

$('2', 0), ('0', 1), ('12', 2), ('02', 3), ('12', 4), ('01', 5), ('2', 6), ('2', 7)$

$$Output \text{ (For ID)} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Algorithm 5 Floyd-Warshall Pseudocode for Name or ID

Weight or distance matrix of mapped (character,index) pair

Output Distance matrix of Name or ID

Example Input:SATYAM

Mapped Input: $(S, 0), (A, 1), (T, 2), (Y, 3), (A, 4), (M, 5)$

```
1: function FLOYDWARSHALL( $W[1..V][1..V]$ )           ▶  $W$  is the weight
   matrix of  $V \times V$  dimension
2:    $D \leftarrow W$                                    ▶  $D$  is initial weight matrix of starting graph
3:   for  $k \leftarrow 1$  to  $V$  do
4:     for  $i \leftarrow 1$  to  $V$  do
5:       for  $j \leftarrow 1$  to  $V$  do
6:          $D^k(i, j) \leftarrow \min\{R^{k-1}(i, j), R^{k-1}(i, k) + R^{k-1}(k, j)\}$ 
```
