Question-Answers

1. **Write down DFS algorithm and analyse the time complexity. What are different classification of edges that can be encountered during DFS operation and how it is classified?**

(Write the algorithm for DFS and analyse its complexity. –May 2019-4 mark, What are different classification of edges that can be encountered during DFS operation and how it is classified? Explain with example –December 2019-4mark)

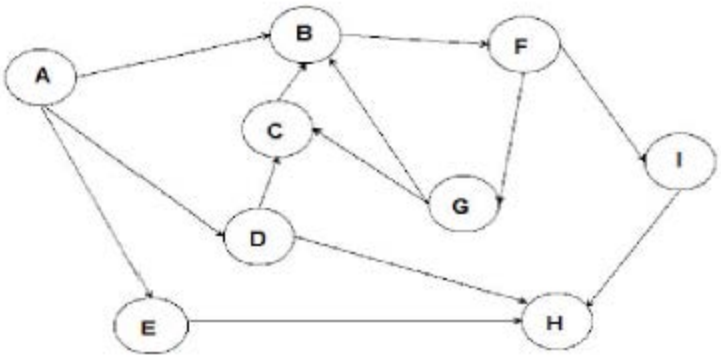
```
DFS(G)
1  for each vertex u ∈ G.V
2      u.color = WHITE
3      u.π = NIL
4  time = 0
5  for each vertex u ∈ G.V
6      if u.color == WHITE
7          DFS-VISIT(G, u)
```

```
DFS-VISIT(G, u)
1  time = time + 1
2  u.d = time
3  u.color = GRAY
4  for each v ∈ G.Adj[u]
5      if v.color == WHITE
6          v.π = u
7          DFS-VISIT(G, v)
8  u.color = BLACK
9  time = time + 1
10 u.f = time
```

The loops on lines 1-3 and lines 5-7 of DFS take time θ(V), exclusive of the time to execute the calls to DFS-VISIT.The procedure DFS-VISIT is called exactly once for each vertex vεV , since the vertex u on which DFS-VISIT is invoked must be white and the first thing DFS-VISIT does is paint vertex u gray.The total cost of executing lines 4-7 of DFS-VISIT is θ(E). The running time of DFS is therefore θ(V + E).
Algorithm – 2 marks, complexity analysis – 2 marks.

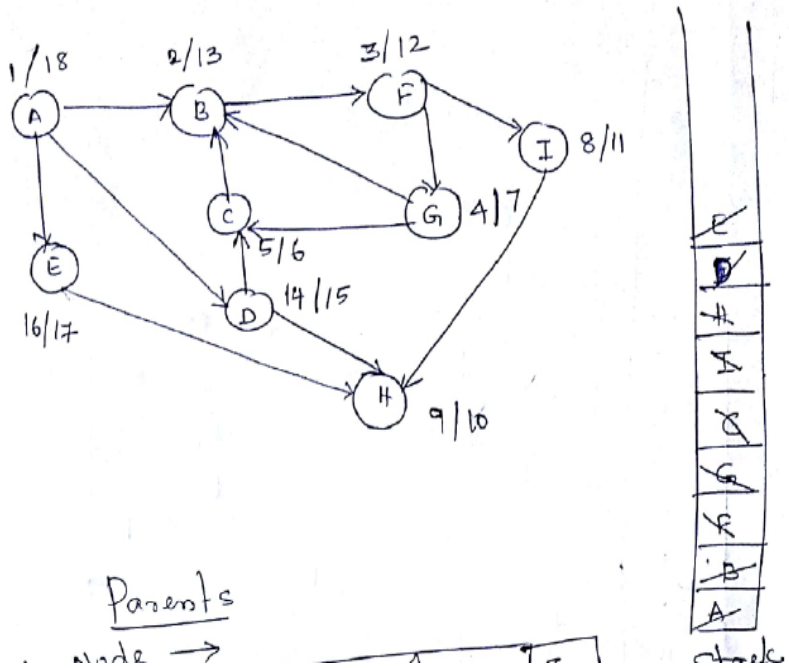
classification of edges :
a) Tree edges - edge (u,v) is a tree edge if v was first discovered by exploring edge (u,v).
b) Back edges - edge (u,v) connecting a vertex u to an ancestor u in a depthfirst tree.
c) Forward edges - edge (u,v) connecting a vertex u to a descendant v in a depth-first tree.
d) Cross edges are all other edges

2. **Perform DFS traversal on the above graph starting from node A. Where multiple node choices may be available for next travel, choose the next node in alphabetical order. Classify the edges of the graph into different category.**



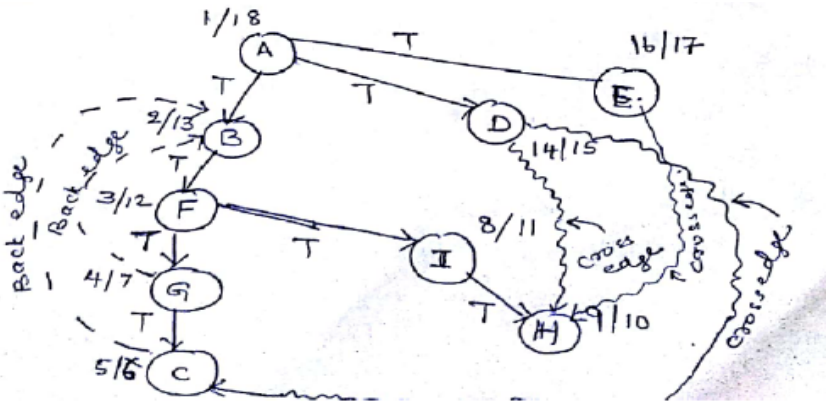
Answer:

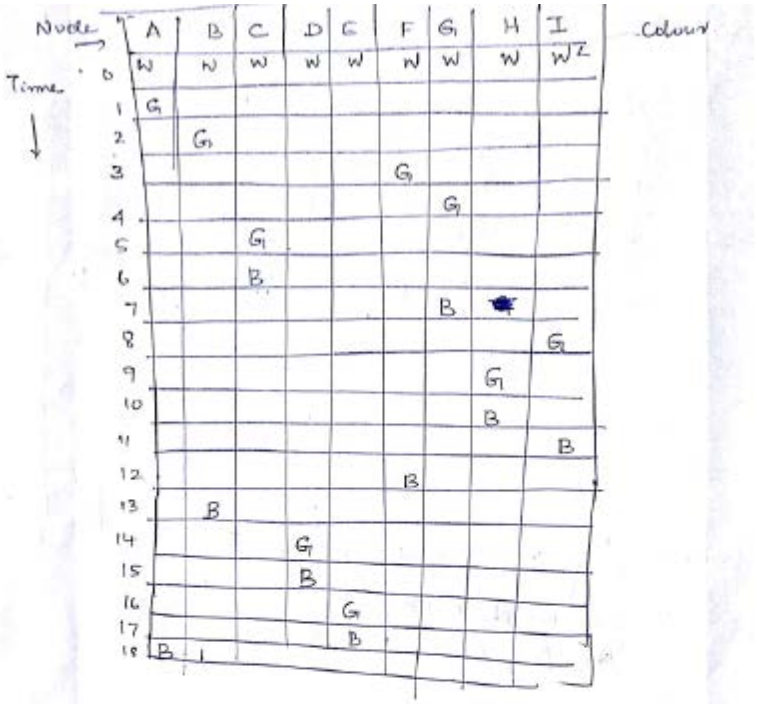
DFS Search starting from Node A



Parents

Node →





(u,v) Tree edges · ($d(u) < d(v) < f(v) < f(u)$)

AB, BF, FG, GC, FI, IH, AD, AE

(u,v) ∈ Back edge · ($d(v) < d(u) < f(u) < f(v)$)

CB, GB ·

(u,v) ∈ Cross edges ·

DC, DH, EH,