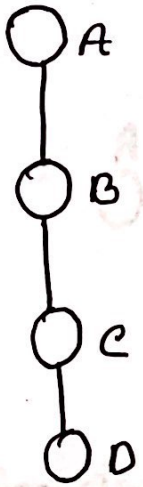
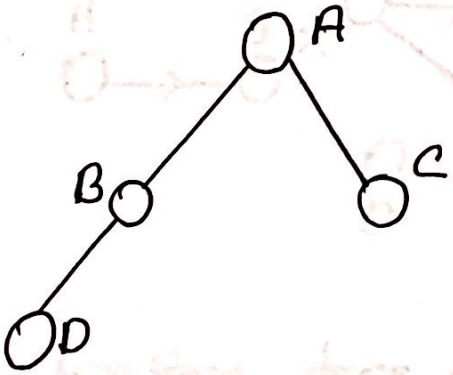
Task-1



DFS : A B C D

BFS : A B C D

So, DFS And BFS of this graph are same.



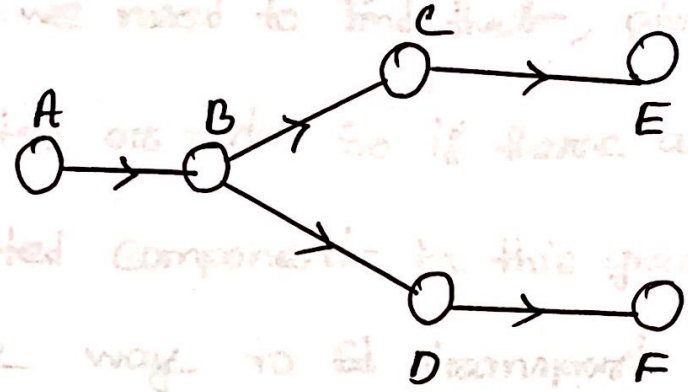
DFS: ~~A B C D~~ A B D C

BFS: A B C D

So, DFS and BFS of this graph is different

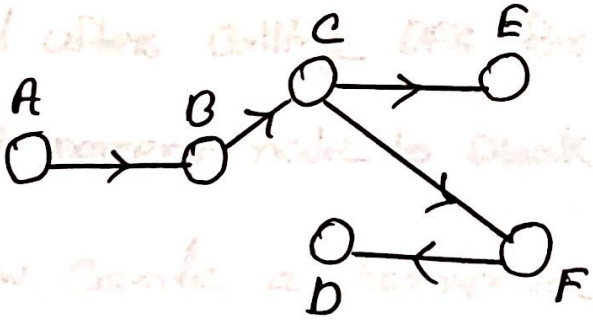
Task:-2

BFS: A B C D E F



Resulting tree of BFS

DFS: A B C E F D



Resulting tree of DFS

Task:-3

We can assume that this is directed graph. So just simply we need to find that, given graph is Strongly Connected or not! So if there are more than 1 Strongly Connected component in this graph then there is no possible way to st transport fuel from any node to any other node in the graph.

Algorithm:-

Create an empty stack and Do dfs on that graph. And after calling DFS for all its adjacent child node add parent node to stack.

Now Create a transpose directed graph. where if in main graph there are a edge A to B, then in transpose graph there will be a edge B to A. (we need to just reverse the direction).

Now pop a node from Stack. let's call it V. Now take V as a root and Do bfs. Now if we are able to visit all nodes in graph then there is a way to transport fuel from any node to any other node in graph. If we are not able to visit all node in that graph then there no possible way to transpose fuel from any node to any other node in the graph.

Why this works:- This is a Kosaraju's Algorith. The number of connected component in transpose graph is same as the numbers of connected component in main graph.

Time complexity:- We are using only Two bfs. And running time of Bfs in O(n+m). So running time of this Algo is O (n+m).

Task:-4

We can solve this using just a simple bfs and dfs.

Algorith:

Create a array named connected. where for every index i in connected [i] there will be value of how many building are connected with i.

then we will do a loop from 1 to n for every powers plants lets call. lets call coorrent power plants is v.

Now we need to find how many building's are connected with v. we can find it just using a simple bfs and dfs.

take v as a root and do bfs or dfs and count the number of buildings are connected with v. And save that value in connected [v].

Now just find the maximum value index in connected array, let call it s.

So DPDC should install the Generators in S and that will provide back up power to the most buildings upon plant failure

Why this Algorith:— So we are finding how ~~an~~ many buildings are connected with each powers plants, And Choose a powers plants with maximum connected buildings for install generators. we are finding all possible way and take optimal. so that Algorith should work.

Running time:— First thing first, what is the maximum number of edges are possible? there are n^3 buildings, So maximum number of edges can be $((n^3 \times (n^3-1))/2)+1$ (1 for powers plant), lets call it E and maximum number of node can be $(n+n^3)$, let's call it V.

now we can't visit a node more than once. So time complexity is similar as bfs and dfs.

So overall time complexity is $O(E*V) = O(((n^3 \times (n^3-1))/2)+1 + (n+n^3))$. which is less than ~~O(n)~~ $O(n^6)$.

Task:- 5

a) Topological ordering of G is 420135

number of distinct topological ordering G is : 8

And topological orderings are:

420135
420153
420315
420351
420513
420531
421035
421053

b) if we ~~need~~ add edge 5 to 4 in graph G then G will be Simple Graph with no topological ordering.

There are 4 distinct single edge that could be added to G to Construct a simple graph with no to topological ordering:
And that edges are:-

0 to 4	5 to 2
3 to 2	5 to 4