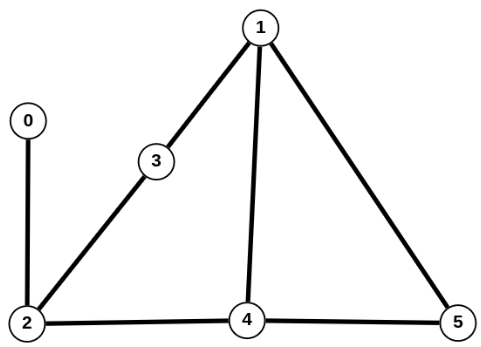
Problem Set 5

Name: Your Name

Collaborators: Name1, Name2

Problem 5-1.

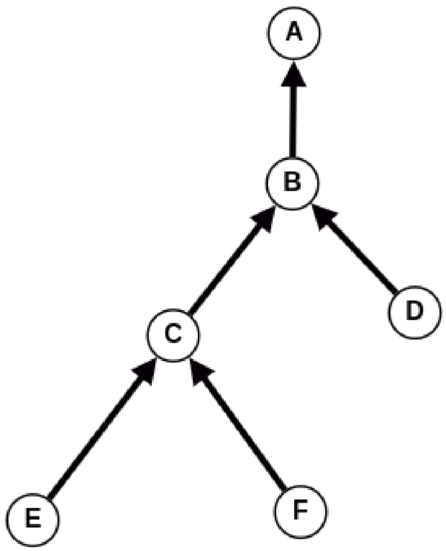


(a)

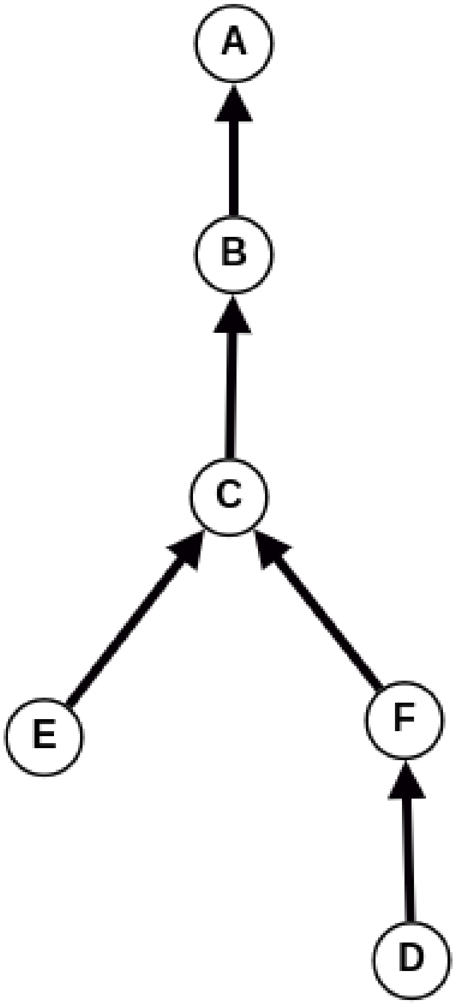
(b)

```
Adj ={
  "A" : ["B"],
  "B" : ["C", "D"],
  "C" : ["E", "F"],
  "D" : ["E", "F"],
  "E" : [],
  "F" : ["E", "D"],
}
```

(c) BFS : [A, B, C, D, E, F]



DFS : [A, B, C, E, F, D]



(d) we can remove **(F, D)** which gives a **DAG** with a topological order of **[A, B, C, D, F, E]**
and **(D, F)** which gives a **DAG** with a topological order of **[A, B, C, F, D, E]**

Problem 5-2. Construct a graph that contains the buildings and plants as its vertices and the edges are the wires in $O(n^4)$

Problem 5-3. Construct a graph $G = (V, E)$, V is the set of friends which every two short-circuiting friends are connected by a single an edge in E. to make this party possible we need G to be a bipartite graph. if G is a bipartite graph, then:

- G is 2-colorable.
- G does not contain any cycles of odd length.

The following algorithm will do the job:

- run **Full-BFS** on G and color evey vertex at even level with the same color of the source vertex.
- check every edge in E to see if it connects two vertices with the same color, if so then G is not a bipartite graph and this party is not possible, else G is a bipartite graph.

Time complexity: O(n)

- Constructing G is $O(n)$, because we have n edges, and vertices are $O(n)$ because it is at most $2n$ vertices.
- Running Full-BFS is $O(n)$, since it is linear in the graph size which is $O(n)$.
- Checking every edge is $O(n)$.

Problem 5-4.

Problem 5-5.

Problem 5-6.

(a)

(b)

(c)

(d) Submit your implementation to `alg.mit.edu`.