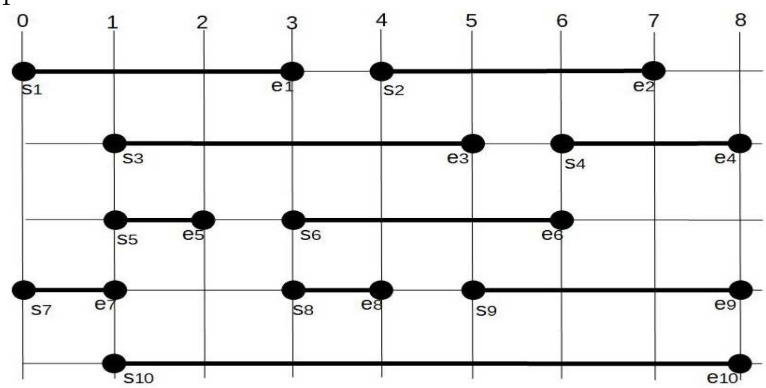
1. An undirected graph $G = (V, E)$ is bipartite if the set of vertices can be partitioned into two sets V_1 and V_2 such that $\forall(u, v) \in E$, $(u \in V_1 \ \wedge \ v \in V_2)$.
   (a) Give a linear time algorithm that determines if a graph is bipartite.
   (b) Explain why your algorithm is correct
2. Give a formal proof that a digraph $G = (V, E)$ has a cycle if and only if DFS finds a back edge.
3. Given the following adjacency list for G:
   y→x; z→w,y; s→z,w; t→v,u; x→z; w→x; v→w,s; u→v
   (a) Give the adjacency matrix.
   (b) Draw the graph
   (c) Assume that explore and DFS **process vertices in lexicographical order**. Find the pre and post visit numbers of each vertex.
   (d) For each vertex indicate its type (tree, forward, back, or cross)
4. Prove that every DAG has at least one sink and at least one source.
5. Let $G = (V, E)$ be an undirected weighted graph.
   (a) Prove that if there are two distinct MST T_1, T_2 then G has at least two edges with the same weight.
   (b) Prove that if all the edges of G have distinct weights then the MST is unique.
6. Given a set of n activities with starting time s_i and end time e_i:

$$(s_1, e_1), (s_2, e_2), \ldots, (s_n, e_n)$$

   Here is an example where $n = 10$



   (a) Design a greedy algorithm to find a maximum set of non-overlapping activities.
   (b) What is the running time of your algorithm?
   (c) What would you need to prove to show that your algorithm is correct (no need to write the proof).
   (d) What is the maximum set of non-overlapping activities given by your algorithm on the example given above?

7. Given the directed weighted graph described by the following edge list (the last column is the weight)

edge	weight
(A,B)	4
(A,C)	2
(B,C)	3
(B,D)	2
(B,E)	3
(C,B)	1
(C,D)	4
(C,E)	5
(E,D)	1

   (a) Draw the graph
   (b) Trace Dijkstra's algorithm on the graph starting from vertex A, showing the distance for each vertex for every step. PROCESS VERTICES ALPHABETICALLY.
   (c) Draw the final shortest path tree

8. Give an example of an undirected weighted graph where a MST is not equal to the tree of shortest paths from a given source vertex.

9. Given an undirected weighted graph $G = (V, E)$, $w : E \to (0, \infty)$
   (a) Show that there is a MST of G that does not contain the heaviest edge in a cycle.
   (b) Prove that the following MST algorithm is correct:

       sort the edges according to their weights
       for each edge $e \in E$, in decreasing order of w
           if e is part of a cycle of G
               $G = (V, E - e)$
       return G

   (c) On each iteration, the algorithm must check whether there is a cycle containing a specific edge e. Give a linear-time algorithm for this task, and justify its correctness.
   (d) What is the time complexity of the MST algorithm from part (b) using (c)?