Questions

Here, we assume that the graph has no loop as well as no parallel edges until specified.

1 Graph Representation

- Given an adjacency-list representation of a graph $G = (V, E)$, write an algorithm to obtain the adjacency-matrix representation of G. Also, obtain the time complexity of this algorithm.
- Given an adjacency-matrix representation of a graph $G = (V, E)$, write an algorithm to obtain the adjacency-list representation of G. Also, obtain the time complexity of this algorithm.
- Given an adjacency-list representation of a graph $G = (V, E)$, write an algorithm to obtain the adjacency-list representation of the transpose of G, *i.e.*, G^T. Also, obtain the time complexity of this algorithm.
- Given an adjacency-matrix representation of a graph $G = (V, E)$, write an algorithm to obtain the adjacency-matrix representation of the transpose of G, *i.e.*, G^T. Also, obtain the time complexity of this algorithm.

2 Degree

- Given an adjacency-list representation of a directed graph $G = (V, E)$, write an algorithm to obtain the in-degree and out-degree of all the vertices. Also, obtain the time complexity of this algorithm.
- Given an adjacency-matrix representation of a directed graph $G = (V, E)$, write an algorithm to obtain the in-degree and out-degree of all the vertices. Also, obtain the time complexity of this algorithm.
- Given an adjacency-list representation of an undirected graph $G = (V, E)$, write an algorithm to obtain the degree of all the vertices. Also, obtain the time complexity of this algorithm.
- Given an adjacency-matrix representation of an undirected graph $G = (V, E)$, write an algorithm to obtain the degree of all the vertices. Also, obtain the time complexity of this algorithm.
- Given an undirected graph $G = (V, E)$, what is the sum of the degree of all the vertices?
- Given a directed graph $G = (V, E)$, what is the sum of the in-degree and out-degree of all the vertices?
- What can be the maximum and minimum degree of a node in case of an undirected graph?
- What can be the maximum and minimum in-degree/out-degree of a node in case of a directed graph?

3 True/False with Reasoning

- Every connected undirected graph is complete?
- Every complete undirected graph is connected?
- Every strongly connected graph is weakly connected graph?
- Every weakly connected graph is strongly connected graph?

4 BFS

- Time complexity of BFS traversal when the graph is represented as –
  - Adjacency-list
  - Adjacency-matrix
- Will the queue used in BFS traversal contain the same vertex more than once? Why.
- What happen if we do not check whether a **vertex has been visited or not** before inserting the vertex into the queue.
  - For directed graph
  - For undirected graph

5 DFS

- Given a directed graph $G = (V, E)$, check whether this graph G is strongly connected or not.

- Given a directed graph $G = (V, E)$, check whether this graph G contains a cycle or not.
- Given a directed graph $G = (V, E)$, arrange the vertices of this graph is such a manner the if there is an edge $(u, v) \in G.E$, then u should come before v in the arrangement.
- Given an undirected graph $G = (V, E)$, check whether this graph G contains a cycle or not.
- Given an undirected graph $G = (V, E)$, check whether this graph G is 2-edge connected or not.
- Given an undirected graph $G = (V, E)$, obtain all the bridges of this graph G.
- Given a directed graph $G = (V, E)$, obtain all the tree edges, forward edges, back edges and cross edges. Obtain the time complexity of this process.

6 Strongly Connected Components

- Given a directed graph $G = (V, E)$. Obtain all the strongly connected components in this graph G *without using* DFS traversal. Write the algorithm for this and analyze the running time.
- What is the maximum and the minimum number of strongly connected components in a directed graph $G = (V, E)$ and why?
- Given a directed graph $G = (V, E)$. Let the number of strongly connected component in this graph G be n where $n = |V|$. Obtain the minimum/maximum number of edges in graph G.
- Given a directed graph $G = (V, E)$ with self loop. Let the number of strongly connected component in this graph G be n where $n = |V|$. Obtain the minimum/maximum number of edges in graph G.
- Given a directed graph $G = (V, E)$. Suppose that G has k strongly connected components $C_1, C_2, \ldots, C_k$. The component graph $G^{SCC} = (V^{SCC}, E^{SCC})$. The vertex set of this component graph V^{SCC} is $v_1, v_2, \ldots, v_k$, and it contains a vertex v_i for each strongly connected component C_i of G. There is an edge $(v_i, v_j) \in E^{SCC}$ if G contains an edge (x, y) for some $x \in C_i$ and some $y \in C_j$. The component graph G^{SCC} is a directed acyclic graph. Why?
- Let C and C' be distinct strongly connected components in directed graph $G = (V, E)$. Let $u.f$ represents the finishing time of a vertex $u \in V$. $f(C)$ and $f(C')$ are the latest finishing time of any vertex in C and C' respectively, *i.e.*, $f(C) = \max_{u \in C} \{u.f\}$ and $f(C') = \max_{v \in C'} \{v.f\}$. Suppose that there is an edge $(u, v) \in E$, where $u \in C$ and $v \in C'$. Then prove that $f(C) > f(C')$.

7 Components: Numerical Questions

- Given an undirected graph $G = (V, E)$. Obtain the maximum/minimum number of components in the graph G.
- Given an undirected graph $G = (V, E)$. Obtain the maximum/minimum number of edges in the graph G when the number of components is k where $1 \leq k \leq |G.V|$.
- Given a directed graph $G = (V, E)$. Obtain the maximum/minimum number of strongly connected components in the graph G.
- Given a directed graph $G = (V, E)$. Obtain the maximum/minimum number of edges in the graph G when the number of strongly connected components is k where $1 \leq k \leq |G.V|$.

8 Minimum Spanning Tree: Kruskal's Algorithm

- Given a connected undirected graph $G = (V, E)$, find the number of edges in
  - Spanning tree
  - Minimum spanning tree
- Show that there is a unique minimum spanning tree if all edges have different costs
- Given a connected undirected graph $G = (V, E)$, what is the number of minimum spanning tree when all edge weights are distinct and why?
- Time complexity of Kruskal's algorithm when it is implemented using
  - Doubly linked-list with head
  - Doubly linked-list with head and tail
  - Doubly linked-list with head and representative

  - Doubly linked-list with head, tail and representative
  - Tree with union by rank/height

9 Minimum Spanning Tree: Prim's Algorithm

- Why is it necessary to have at-least one edge of the cut in spanning tree?
- Time complexity of Prims's algorithm when it is implemented using
  - Min-Heap is used to store the keys of vertices
  - Array is used to store the keys of vertices
  - Linked-list is used to store the keys of vertices
  - Collect all edges of the cut and finding the minimum and create tree

10 Shortest Path

- Why the shorted path should be simple? What will happen if it is not simple.
- Time complexity of Dijkstra's algorithm when $\overline{S}$ is implemented using
  - Min-Heap
  - Array
  - Linked-list
- Maximum/Minimum number of edges in the shortest path from u to v where $u \neq v$. Assume all edge weight are positive.
- Maximum/Minimum number of edges in the shortest path from u to v where $u \neq v$. Assume edge weight can be negative.
  - No negative weight cycle
  - Negative weight cycle reachable from u
  - Negative weight cycle not reachable from u
- Given a weighted directed graph $G = (V, E)$. Write an algorithm to
  - Detect a cycle (if any) and obtain the time complexity
  - Detect a negative weight cycle (if any) and obtain the time complexity
- Given a weighted directed acyclic graph $G = (V, E)$. Write an algorithm to find the shortest path between a source vertex s and all other remaining vertices, and obtain the time complexity.
- In case of Bellman-Ford Algorithm, relaxing an edge more than $n - 1$ times will give you any advantage/disadvantage? Explain.