CS201 Assignment 2: The Concept of Graphs

Maximum Marks: 100

We have discussed the concept of graphs in the class, focusing on simple, undirected graphs. Another important class of graphs is that of *directed graphs* $G = (V, E)$ with $E \subseteq V \times V$ not nessasarily symmetric. In other words, edges (u, v) and (v, u) are distinct from each other. For directed graphs, some of the properties undergo a change. For example, instead of degree of a vertex, there are now two associated numbers: *indegree* and *outdegree* of a vertex. Indegree of v is the number of vertices u such that $(u, v) \in E$ and outdegree of v is the number of vertices u such that $(v, u) \in E$. A *path* from vertex u to vertex v in a directed graph is a sequence of vertices $u_0 = u$, u_1, ..., $u_k = v$ such that edge $(u_i, u_{i+1}) \in E$ for $0 \leq i < k$.

- Give example of a directed graph containing vertex u and v such that there is no path from vertex u to v or vice versa, however, there is a path from u to v if we ignore the edge directions. [5 marks]

We say that a directed graph is *connected* if the graph is connected if we ignore the edge directions. We say that a directed graph is *strongly connected* if there is a path from every vertex u to every other vertex v.

- Define relation R on vertices of a directed graph as: uRv if there is a path from u to v and vice versa. Show that R is an equivalence relation in V. [5 marks]

- Relation R splits V into equivalence classes called *strongly connected components*. Consider two equivalence classes V_1 and V_2. Show that all the edges from V_1 to V_2 are in the same direction, i.e., either all are from vertices in V_1 to vertices in V_2, or all are from vertices in V_2 to vertices in V_1. [5 marks]

Trees were defined in the class for undirected graphs. A directed graph is a *tree* if the graph is connected and every vertex of the graph, except one, has indegree exactly one.

- Show that for a directed tree $|E| = |V| - 1$, and for any two vertices u and v, $u \neq v$, if there is a path from u to v then there is no path from v to u. [5 marks]

For a directed tree, the vertex with zero indegree is called *root* of the tree and vertices of zero outdegree are called *leaves* of the tree. For any edge (u, v), vertex v is called *child* of u and u is called *parent* of v. A special type of directed trees are *binary trees* where outdegree of every vertex is at most two. So a vertex

u in a binary tree has at most two children. A binary tree can be drawn on a sheet of paper with root on top, its children below it with one on left side (*left child*) and other on right side (*right child*) (if there are two children). If there is only one child, then it can be drawn either on left side or on right side. How may distinct such drawings of binary trees on n vertices exist? Let T_n be this number.

- Derive a recurrance relation for T_n. [10 marks]
- Derive a formula for T_n using generating functions. [15 marks]

Consider a directed graph $G = (V, E)$. Let V_1, V_2, ..., V_k be its strongly connected components. Define graph $H = (V_H, E_H)$ as: $V_H = \{1, 2, \ldots, k\}$, and edge $(i, j) \in E_H$ iff there is an edge from a vertex in V_i to a vertex in V_j.

- Prove that H is a directed tree. [5 marks]

Consider a graph $G = (V, E)$, directed or undirected. A *cycle* of G is a path v_1, v_2, ..., v_k such that $v_k = v_1$.

- Prove that an undirected graph is a tree if and only if it is connected and does not have any cycle. [5 marks]
- Prove that for a directed graph, all vertices in a cycle of the graph lie in the same strongly connected component. [5 marks]

A *subgraph* of graph $G = (V, E)$ is a graph (V, E') with $E' \subseteq E$. It is obtained by removing some edges from G. A *spanning tree* of a graph $G = (V, E)$ is a subgraph of G that is a tree.

- Prove that an undirected graph has a spanning tree iff it is connected. [5 marks]

A *weighted* graph is a graph $G = (V, E, w)$ where V and E is the set of vertices and edges as usual, and $w : E \mapsto \mathbb{R}$ is a weight assignment to all edges of the graph. The weight of spanning tree of an undirected, weighted graph equals the sum of the weights of edges in the spanning tree. This allows one to define the notion of *minimum weight spanning tree* of a weighted graph as a spanning tree whose weight is least amongst all spanning trees of the graph.

A minimum spanning tree is very useful in applications. For example, consider the problem of laying water distribution network in a city. Every house in the city is required to be connected. Pipes can be laid between any two houses, with the lengh of pipe being equal to distance between the two houses. This is modeled as creating a weighted graph with vertices prepresenting all houses, edges representing pipes between houses, and weight of an edge equal to the distance between the houses. The water network must be a connected subgraph of this graph. Defining weight of a subgraph to be equal to sum of weights of edges in the subgraph, one would want a minimum weight connected subgraph so that length of pipes used is minimized.

- Prove that minimum weight subgraph is a minimum spanning tree of the graph. [5 marks]

There is a very efficient algorithm to find a minimum spanning tree of a given weighted graph (V, E, w):

```
1. Start with $U = \{v\}$, $v \in V$.
2. If $U = V$, stop.
3. Otherwise, choose the minimum weight edge $(v_1, v_2) in E$ such that
   $v_1 \in U$ and $v_2 \notin U$.
4. Add edge $(v_1, v_2)$ in the spanning tree and $v_2$ to $U$.
5. Go to step 2.
```

- Prove that the spanning tree constructed by the above algorithm is a minim spanning tree. [10 marks]

Another utility of minimum spanning tree is in finding a good solution of *Travelling Salesman Problem* (TSP). The problem is that a salesman needs to visit a given number of cities using the road network and return at the starting city. He obviously wishes to travel minimum possible distance. This is modeled as a weighted graph with vertices representing cities, edges roads between cities, and weight of an edge being lenght of the road. Finding the minimum length tour is known to be a very hard problem to solve, and no efficient solution for this exists. A minimum spanning tree provides a good solution for the problem.

- Prove that weight of a minimum spanning tree of the graph is at most the minimum length tour. [5 marks]

- Prove that minimum length tour is at most twice the weight of a minimum spanning tree. [15 marks]

Therefore, a tour can be created using minimum spanning tree whose length is within a factor of two of minimum length tour.