Due: Nov 22, 2016, in class.

1. (30 points) Please show the progress of building the minimum spanning tree (MST) of the graph in Fig. 1 by Kruskal’s algorithm.

![Figure 1: The weight of each edge is labeled.](image)

2. (10 points) Prove that if all the weights in a graph G are distinct, then G has a unique MST.

3. (20 points) Let the length of a path in a weighted graph be the sum of the weights of the edges in the path. Suppose we want to find a shortest path from vertex a to vertex d by the following greedy strategy: starting from vertex a, we move to its neighbor who has the edge with the smallest weight among all the neighbors. For example, in the following graph Fig. 2, we want to compute the shortest path from a to d. The resulting path by the greedy strategy is $a \rightarrow b \rightarrow c \rightarrow d$ which obviously is not the shortest path. Why does this greedy strategy fail? Specifically, do the two properties of greedy algorithms hold in this case? Please explain the reason.

![Figure 2: The weight of each edge is labeled.](image)

4. (20 points) We use greedy strategy to solve the activity selection problem: each time we select the activity with the smallest finishing time and remove the activities which have overlap with it, then reduce the problem to the sub-problem for the remaining activities. But if we have two CPUs instead, and we still want to select as many as possible activities to process by these two CPUs, does this greedy strategy still work? Similar to question 3, please explain your reason.
5. (20 points) Given a weighted graph, we can use kruskal’s algorithm to compute its MST within $\theta(m \log n)$ time, where $m$ is the number of edges and $n$ is the number of vertices. But consider a special case: there are only two possible weights for all the edges (e.g., the weight could be either 1 or 2), can we design a linear time (i.e., $\theta(n + m)$) algorithm for MST? (hint: you may consider BFS).

6. (bonus for extra 20 points) Consider a generalization of the above question 5: there are $t \geq 2$ possible weights for all the edges (e.g., the weight could be 1, 2, ..., $t$), and we assume $t$ is a small constant. Can we design a linear time (i.e., $\theta(n + m)$) algorithm for MST?