CS 4820, Spring 2018 **Homework 3, Problem 1**
Name: Rongguang Wang
NetID: rw564
Collaborators: Siyuao Liu (sl2928); Yihao Chen (yc2288)

(1) *(10 points)*
Suppose we are given a graph $G = (V, E)$ with costs $(c_e)_{e \in E}$ on the edges and costs $(c_v)_{v \in V}$ on the vertices. We wish to designate a subset of the vertices as *active* and a subset of the edges as *eligible*, such that every inactive vertex can be joined to at least one active vertex by a path made up of eligible edges. Design an algorithm to choose the set of active vertices, A, and the set of eligible edges, F, so as to minimize their combined cost, $\sum_{v \in A} c_v + \sum_{e \in F} c_e$.

Remark: You may assume that G is a connected graph. In particular, if G has n vertices and m edges, you may assume $m \geq n - 1$.

Solution:

The problem is aiming to design a algorithm that select active vertices and eligible edges such that every inactive vertices are connected to the active vertices though the eligible edges from the given graph. The sum of the weight of both active vertices and the eligible edges should be minimized. The problem can be transformed to minimum spanning tree problem. The only difference between the two problem is that the minimum spanning tree does not consider the weight of the vertices. So the problem can be reduced to the MST problem by converting the weight of the vertices to the weight of edges. A virtual vertex should be created and connect to each vertices in the given graph though several newly created edges. Transfer the weight of the vertices to the weight of the newly created edges. Then, the problem was reduced to a MST problem.

The basic algorithm is as following:
1) Transform the original graph by adding edges between each vertex and a new created vertex
2) Transfer the weight of each vertex to the newly added edge connected to each vertex
3) Find the minimum spanning tree using Kruskals Algorithm
4) Transform the MST back to the original graph by converting the edges connected to the newly created vertex back to the vertex
5) Transfer the weight of the newly created edges back to the weight of the vertices they connected to
6) The vertices connected to the newly created edges are selected as the active vertices and the rest vertices are inactive vertices; The edges of the MST are eligible edges

Algorithm's Correctness:

Lemma 1: The transform from the original graph to the reduced MST problem and the transform back to the original problem have no cost.

Suppose the optimal solution of the original problem is set M. Create a virtual vertex connects all the active vertices in M with edge weights as the costs of the vertices they connected to. This is the solution N that the result of the Kruskals Algorithm performing on the reduced problem (virtual vertex created). Since solution M contains no cycles, solution N contains no cycles. Let the optimal solution of the MST problem as N'. Then, the cost of N' is less than the cost of N. Active the vertices that connected to the virtual vertex in set N' with weight that the virtual edges the vertices connected to. Delete the virtual vertex and the edges connected to it. Inactive the vertices that did not connected to the virtual edges. This the solution M' to the original problem. Since all the inactive vertices are connected to the virtual vertex trough the active vertices, the cost of solution M is less than the cost of solution M'. Therefore, the optimal solution of the original problem is the same optimal solution of the transformed MST problem in terms of the cost.

Lemma 2: Every inactive vertex is connected to the active vertex though the eligible edges.

According to the reduction of the original problem, all the vertices are connected to a virtual vertex though a branch of virtual edges. The MST algorithm will connect all vertices including the virtual vertex with no cycle. Thus, there should be at least one virtual edge that connected to the virtual vertex been selected, which means that there should be at least one vertex that connected to the virtual vertex will be activated when converting back to the original problem. In this case, all the non-virtual vertices are connected in the MST through eligible edges and theres one active vertex among them. The rest of the vertices that does not connected to the virtual vertex are inactive vertices. Therefore, every inactive vertex is connected to the active vertex through the eligible edges.

Running Time Analysis:

Tie breaking:
Add a tiny to each of the tied values of the weights of the vertices and the edges. Then, the ties were broken and all vertices and edges are distinct, although some are very close. Now the distinct-only algorithm can be used to solve the problem. The original values of the weights of the vertices and the edges can be recovered after the algorithm was performed.

The time spend in this algorithm consists of: transform the original graph to the graph that has weight only on edges, Kruskals Algorithm which create the minimum spanning tree and transform the MST result back to the graph that has weights on both edges and vertices. There is nearly no cost for the problem reduction since only a new vertex should be created. Similarly, the cost of transform the MST back to the graph is negligible. The only cost comes from the Kruskals Algorithm with n vertices and m edges. Therefore, it spend $(m \log n)$ time.