# Homework 1
## Greedy Algorithms and Discretization

**Problem 1-1.** Consider the following generalization of the minimum cut problem[1]. We are given a graph $G(V, E)$ with edge weights $w(e)$. We are also given a set of terminals $T = \{t_1, t_2, \ldots, t_k\} \subseteq V$. The goal is to find the minimum weight subset of edges to remove so that none of the terminals are connected to each other. In other words, each terminal is in a different connected component.

Consider the following greedy algorithm: Let $C_i$ denote the minimum cost set of edges to delete to disconnect $t_i$ from all the vertices in $T - \{t_i\}$. Show how $C_i$ can be computed by an application of the minimum cut algorithm. We output $C = \cup_{i=1}^{k} C_i$ as our final answer. Show that deleting the edges in $C$ actually disconnects all the terminals. We will now show that $C$ is a 2-approximation to the optimal solution.

Let $O$ denote the set of edges deleted by the optimal solution. This results in at least $k$ connected components, each containing one of the terminals. Consider the subset of deleted edges $O_i \subseteq O$ that is incident on the component containing $t_i$. Note that deleting just the set $O_i$ disconnects $t_i$ from $T - \{t_i\}$. Therefore, show that $w(O_i) \geq w(C_i)$ (where $w(S)$ is the weight of the edges in set $S$).

Show that each edge in $O$ appears in at most two of the subsets $O_1, O_2, \ldots, O_k$. Therefore, $2w(O) \geq \sum_{i=1}^{k} w(O_i)$. Finally, show that $w(C) \leq 2w(O)$.

**Problem 1-2.** Consider the following algorithm for knapsack. We sort the objects in decreasing order of profit to sizei ratio. We greedily pick objects until the total size we select exceeds the capacity of the knapsack. Let $S$ be the set of chosen objects, and let $i$ be the index of the last object we picked. Let $B$ denote the capacity of the knapsack. Note that size$(S) > B$, so $S$ is not a feasible solution.

Show that profit$(S) \geq OPT$, where $OPT$ is the optimal profit. Now consider the two sets $S_1 = S - \{i\}$ and $S_2 = \{i\}$. If profit$(S_1) >$ profit$(S_2)$, we output $S_1$ as our final solution,

[1] If you are not familiar with the minimum cut problem, please read it up from Cormen, Leiserson and Rivest's book on Algorithms.

else we output $S_2$. Show that the final solution is feasible, that is, its size is at most $B$. Also show that this solution has profit at least $\frac{OPT}{2}$. Construct an example to where the profit is $\frac{OPT}{2}$.

**Problem 1-3.** We are given a graph $G(V, E)$, where each edge has two length functions $c(e)$ and $l(e)$. We are given a source node $s$ and a destination node $t$. Given two numbers $C$ and $L$, the goal is to compute a path from $s$ to $t$ whose length according to the first length function is at most $C$ and whose length in the second function is at most $L$, or output that no such path exists.

Write a dynamic programming pseudo-polynomial time algorithm for this problem. Given any $\epsilon > 0$, show how to compute a path of lengths at most $C(1+\epsilon)$ and $L(1+\epsilon)$ according to the two length functions, in time $poly(|E|, \frac{1}{\epsilon})$, or output that no path of lengths at most $C$ and $L$ exists.

*Hint:* For the dynamic program, let $P(s, v, C', L')$ denote whether there exists a path from $s$ to $v$ whose lengths according to the two length functions are at most $C'$ and $L'$ respectively. For the discretization, use $C$ and $L$ somehow.