# CIS 502 - Algorithms

## Fall 2015 Homework 7 Solutions

**Problem 1** *Problem 37, page 526–527.*

**Solution:** It is easy to see that the problem is in NP. We show that it is NP-Hard.

We will reduce partition to this problem. Given a set of positive integers $\{a_i\}$ partition asks if there is a set $S$ such that $\sum_{i \in S} a_i = \frac{1}{2}\sum_i a_i$. We now produce a graph over the vertices $s = v_0, v_1, \ldots, v_n = t$.

There are two edges from $v_{i-1}$ to $v_i$, one of length $a_i$ and risk 0, and the other of risk $a_i$ and length 0. We set $L = R = \frac{1}{2}\sum_i a_i$.

If there exists a solution for the partition problem, denoted by set $S$, then if we take the first edge out of $v_{i-1}$ if $i \in S$ and the second edge if $i \notin S$. Observe that the length of the path is only determined by the $i \in S$ and the length is $\sum_{i \in S} a_i = \frac{1}{2}\sum_i a_i = L$. The risk is determined by the $i \notin S$ and $\sum_{i \notin S} a_i = \frac{1}{2}\sum_i a_i = R$.

In the reverse direction, if there exists a path of risk and length at most $\frac{1}{2}\sum_i a_i$ each, then first observe that there must exist a simple path (without revisiting any vertex) since risks and lengths are non-negative. Now consider $S'$ to be the set that the path took the first edge out of $v_{i-1}$. Therefore based on the length

$$\sum_{i \in S'} a_i \leq L = \frac{1}{2}\sum_i a_i$$

Now based on the risk,

$$\sum_{i \notin S'} a_i \leq R = \frac{1}{2}\sum_i a_i$$

Both these equations can only be satisfied when $\sum_{i \in S'} a_i = \sum_{i \notin S'} a_i = \frac{1}{2}\sum_i a_i$ which is a solution for the partition problem.

**Problem 2** *Problem 10, page 656–657, of textbook.*

**Solution:** Part (a). Suppose not. Then $v \notin S$ and there does not exist $v' \in S$ such that $w(v) \leq w(v')$ and $(v, v')$ is an edge of $G$. So for all nodes $v' \in S$ either the edge $(v, v')$ is not an edge of $G$ – which means $v$ has not been deleted and can be picked (contradiction). Or $(v, v')$ is an edge of $G$ and $w(v) > w(v')$, but then we should have picked $v$ instead of $v'$ to add to $S$ – again a contradiction.

Part (b). Suppose $T$ is any other independent set. Consider the following accounting process. As we add node $v$ to $S$ we delete $v$ or all of its neighbors from $T$ and charge the weight of the deleted nodes to $v$. The total charge on the elements of $S$ is therefore the total weight of $T$. However a node $v$ picks up a charge at most $w_v$ since it has at most 4 neighbors and each of which have weight at most $w_v$ (otherwise we would picked that neighbor, which would have not yet deleted from $G$). Therefore the total charge is at most 4 times the weight of $S$. Therefore $w(T) = \sum_{u \in T} w(u) \leq 4w(S)$.

**Problem 3** *Problem 11, page 658, of textbook.*

**Solution:** There are two solutions (they are very similar).

Note we can assume that $w_i \leq W$ for each $i$. Consider setting $s_i = \left\lfloor \frac{nw_i}{\epsilon W} \right\rfloor$ and $S = \left\lfloor \frac{n}{\epsilon} \right\rfloor$. Now consider a knapsack with sizes $s_i$ and value $v_i$ (same as before) such that the size does not exceed $S$. If there existed a solution $O$ such that $\sum_{i \in O} w_i \leq W$ and $\sum_{i \in O} v_i \geq V$ then the set $O$ also satisfies

$$\sum_{i \in O} s_i \leq \sum_{i \in O} \left\lfloor \frac{nw_i}{\epsilon W} \right\rfloor \leq \sum_{i \in O} \frac{nw_i}{\epsilon W} \leq \frac{n}{\epsilon} \tag{1}$$

At the same time $\sum_{i \in O} s_i$ is an integer and therefore $\sum_{i \in O} s_i \leq \left\lfloor \frac{n}{\epsilon} \right\rfloor = S$. Therefore by Dynamic programming (using the fact that the sizes are integers and the sum is at most $n/\epsilon$) we can find a set $A$ such that $\sum_{i \in A} s_i \leq S$ and $\sum_{i \in A} v_i \geq V$. Now

$$\sum_{i\in A} w_i = \frac{W\epsilon}{n}\sum_{i\in A}\left(\frac{nw_i}{\epsilon W}\right) \le \frac{W\epsilon}{n}\sum_{i\in A}\left(\left\lfloor\frac{nw_i}{\epsilon W}\right\rfloor + 1\right)$$
$$\le \frac{W\epsilon}{n}\sum_{i\in A}(s_i+1) \le \left(\frac{W\epsilon}{n}\sum_{i\in A} s_i\right) + \frac{W\epsilon}{n}\sum_{i\in A} 1 \le W + \epsilon W$$

which is what was desired.

In the second solution we can set $s_i = \lceil \frac{nw_i}{\epsilon W} \rceil$ and set $S = \lfloor \frac{n}{\epsilon} \rfloor + n$. Equation 1 is now

$$\sum_{i\in O} s_i \le \sum_{i\in O}\left\lceil\frac{nw_i}{\epsilon W}\right\rceil \le \sum_{i\in O}\left(\frac{nw_i}{\epsilon W}+1\right) \le \frac{n}{\epsilon} + n \tag{2}$$

If we now find a set $A$ such that $\sum_i s_i \le \lfloor \frac{n}{\epsilon} \rfloor + n$ and $\sum_{i\in A} v_i \ge V$ then,

$$\sum_{i\in A} w_i = \frac{W\epsilon}{n}\sum_{i\in A}\left(\frac{nw_i}{\epsilon W}\right) \le \frac{W\epsilon}{n}\sum_{i\in A}\left\lceil\frac{nw_i}{\epsilon W}\right\rceil \qquad \le \frac{W\epsilon}{n}\sum_{i\in A} s_i \le \frac{W\epsilon}{n}\left(\left\lfloor\frac{n}{\epsilon}\right\rfloor + n\right) \le W + \epsilon W$$

**Problem 4** *Problem 12, page 658–659, of textbook.*

**Solution:** This problems takes the idea behind set-cover greedy algorithm to the next level. If ever you are faced with a problem to approximate, this is a good analysis to remember. The kernel of the algorithm is that we can find the "best ratio" set and add it to our solution. In set cover the best ratio was minimizing cost of the set divided by the number of uncovered elements. Here the covering is a bit more involved. Suppose that we have an uncovered set of demands $U_i$ after iteration $i$. Suppose the set of facilities bbuilt upto this point is $F_i$. In this case finding the "best ratio" set is complicated. Two main issues arise:

- **The best ratio set corresponds to a new facility**. Suppose we chose facility $j$ and are going to pay cost $f_j$ upfront. Now suppose we cover the elements of set $S(j) \subseteq U_i$; then the ratio is

$$\frac{1}{|S(j)|}\left(f_j + \sum_{k\in S(j)} d_{kj}\right)$$

  So for the same fixed $|S(j)|$ the ratio is smallest if we include the closest $|S(j)|$ points in $U_i$ (otherwise a greedy swap improves the ratio).

- **The best ratio set corresponds to an already constructed facility.** In this case we want to minimize

$$\frac{1}{|S|}\left(\sum_{k\in S}\min_{j\in F_i} d_{kj}\right)$$

  where $k$ attaches to the closest previously built facility. But notice that this ratio is minimized by a single point in $U_i$ closest to $F_i$.

The remainder of the analysis is the same as the set cover charging scheme. Suppose the optimum solution uses some facility $j^*$ and covers $S^*$ elements using $j^*$. Then the total charge to all the elements in $S^*$ is at most

$$(\text{Cost of covering } S^* \text{ by } j^*)\left(\frac{1}{|S^*|} + \frac{1}{|S^*|-1}\cdots + 1\right)$$

which adds up to at most $H(n)$ times the cost of the optimum solution.