Advanced Analysis of Algorithms - Quiz II (Solutions)

L. Kovalchick
LCSEE,
West Virginia University,
Morgantown, WV
{lynn@csee.wvu.edu}

1 Problems

1. Let $G =< V, E, c >$ denote an undirected graph with vertex set V, edge set E and a cost function c that assigns a positive integer $c(e)$ to each edge $e \in E$. If all the edge costs are unique, argue that G has a unique Minimum Spanning Tree. (7 points)

   Proof:
   Suppose that 2 unique Minimum Spanning Trees exist; call them M and M'. Let e_1 be the edge with the smallest cost that belongs to one of the trees, but not both. We will assume, without loss of generality, that e_1 belongs to M. Now, consider the graph G', formed by e_1 and M'. Obviously G' must contain a cycle, since M' was a Minimum Spanning Tree before e_1 was added. Observe that one edge of the cycle in G' must not appear in M (call this edge e_2), otherwise, M would not be a Minimum Spanning Tree (i.e., M would contain a cycle). Observe that e_1 must have a smaller cost than e_2, otherwise, e_2 would have been in M. By deleting e_2 from M' and replacing it with e_1, we have reduced the weight of the spanning tree M'. This is a contradiction since our original assumption was that both trees were Minimum Spanning Trees. □

2. The Hamilton Cycle problem is defined as follows:
   HC: *Given an undirected graph $G =< V, E >$, with vertex set V and edge set E, is there a simple cycle in G, that visits all the vertices in V?*

   The Traveling Salesman problem is defined as follows:
   TSP: *Given a complete undirected graph $G =< V, E, c >$, with vertex set V, edge set E and a cost function c, that assigns a positive cost $c(e)$ to each edge $e \in E$, and a number K, does there exist a tour (simple cycle) of G that visits all the vertices in V, and whose cost at is at most K? The cost of a tour is defined as the sum of the costs of the edges in the tour.*

   Describe a polynomial time transformation of **HC** to **TSP**. (7 points)

   Solution:
   Let $G =< V, E >$ be an instance of the Hamilton Cycle problem. We will construct an instance of the Traveling Salesman problem as follows. We form the complete graph $G' =< V, E' >$. We then define the cost function c as:

$$\begin{aligned} c(u, v) &= 0, \quad \text{if } (u, v) \in \text{the original graph}, \\ &= 1, \quad \text{otherwise.} \end{aligned}$$

   We now use $< G', c, 0 >$ as our instance of the Traveling Salesman problem. Notice that this transformation takes only polynomial time. □

3. Develop a counterexample to show that the greedy algorithm developed for the fractional Knapsack problem does not work for the 0/1 Knapsack problem. (6 points)

Solution:

Recall that the greedy algorithm developed for the fractional Knapsack problem is defined as follows. We first compute the value per pound of each item (i.e., $\frac{v_i}{w_i}$). We then pack as much of the item with the greatest value per pound ratio into our knapsack. If the entire item is packed and there is still room left, we then pack as much of the item with the second greatest value per pound ratio, and so forth until our knapsack is filled to capacity.

In order to create a counterexample for the 0/1 Knapsack problem, consider the following setup. Let S denote our set of items, where $S = \{1, 2, 3\}$. Let the weight of our items be $w_1 = 10$, $w_2 = 20$, and $w_3 = 30$ pounds, and let $M = 50$ denote the total weight that our knapsack can hold. Finally, let each item have a value in dollars of $p_1 = \$60$, $p_2 = \$100$, and $p_3 = \$120$.

If we use the greedy strategy to solve the 0/1 Knapsack problem, we first need to calculate the value per pound ratio of each item; these ratios are $\frac{60}{10} = 6$, $\frac{100}{20} = 5$, and $\frac{120}{30} = 4$, respectively. We then pack the item with the greatest value per pound ratio into our knapsack. After packing the first item, our knapsack is not yet filled to capacity, so we pack the item with the second highest value per pound ratio. Observe that after packing the second item, we still have room left in our knapsack, but we can not pack the third and final item, because there is not sufficient space for the entire item. Adding the values of the two items in our knapsack, we have a total profit of \$160, using the greedy strategy.

Now, consider what happens if we instead take the item with the greatest value first. We still have room in our knapsack; so we will take the item with the second greatest value. After packing these two items, we have reached our capacity and our total profit is \$220, using this strategy.

We have a counterexample showing that the greedy strategy does not produce the optimal solution. Therefore, we conclude that the greedy strategy does not work for the 0/1 Knapsack problem. □