# Greedy Algorithms (Practice Set-1)

(Courtesy Aditya Pancholi)

1. Consider Interval Scheduling Problem where we are given a set of intervals $\mathcal{I} = \{1, 2, \ldots, n\}$ where each $i \in \mathcal{I}$ has starting time $s_i$ and finishing time $f_i$.
   (a) Consider the greedy approach where we include the interval with minimum $s_i$ (first greedy approach discussed in the book).
      1. Give an input instance where the approach will produce an Optimal Solution.
      2. Give an input instance where the approach will produce a solution with intervals exactly half the Optimal Solution.
      3. Give an input instance where the approach will produce a solution with intervals almost $\frac{1}{n}^{th}$ the Optimal Solution.

   (b) Consider the greedy approach where we include the interval with minimum span (i.e. $f_i - s_i$) (second greedy approach discussed in the book).
      1. Give an input instance where the approach will produce an Optimal Solution.
      2. Give an input instance where the approach will produce a solution with intervals exactly half the Optimal Solution.

   (c) Consider the greedy approach where we include the interval with minimum conflicts (third greedy approach discussed in the book).
      1. Give an input instance where the approach will produce an Optimal Solution.
      2. Give an input instance where the approach will produce a solution with intervals exactly half the Optimal Solution.

   (d) Consider the following greedy approach.
   *Approach* : Consider the interval that starts the last (has highest $f_i$). Include the interval in our solution and eliminate all the conflicting intervals. Repeat the process, until we have no intervals left.
      1. Does the above approach produces an optimal solutions? Justify.

   (e) Analyze the running time for each of the above greedy algorithms.

2. Consider $Dijkstra's$ Algorithm.
   (a) Does it produces an optimal solution if we allow some edge weights to be zero? (Edge weight can be either positive or zero). Justify.

   (b) Does it produces an optimal solution if we allow some edge weights to be negative? (Edge weight can be either positive or negative). Justify.

3. For a weighted graph $G(, V, E)$, we all edge weights strictly positive, suppose we modify the definition of shortest path as follows.
   $Definition$ : The length of a path $\mathcal{P}$ of a Graph $G(, V, E)$ is defined as **product** of the edge weights of all the edges in the path $\mathcal{P}$.
   (a) Can the standard $Dijkstra's$ algorithm be applied for the above problem? (Without modifying the Algorithm)
   (b) Design an Algorithm on the similar lines as $Dijkstra's$ to solve the above problem. Give the optimality proof for your algorithm.

4. The following statements may may or may not be correct. In each case, either prove it (if it is correct) or give a counter example (if it is not correct). Always assume that the graph $G(V, E)$ is un-directed and connected. Do not assume the edge weights are distinct unless specifically stated.
   (a) If graph $G$ has more than $|V| - 1$ edges, and there is a unique heaviest edge then the edge cannot be the part of Minimum Spanning Tree.
   (b) If $G$ has a unique heaviest edge $e$, then $e$ cannot be part of any Minimum Spanning Tree.
   (c) If $G$ has a unique heaviest edge $e$, then $e$ cannot be part of any shortest path between a pair of vertices $v_1, v_2$.
   (d) If $e$ be a minimum weight edge, then it must be a part of every MST.
   (e) If $e$ be a minimum weight edge, then it must be a part of some MST.
   (f) If $e$ be a unique minimum weight edge, then it must be a part of every MST.
   (g) The shortest path between two nodes is necessarily the part of every MST.

5. The following statements may may or may not be correct. In each case, either prove it (if it is correct) or give a counter example (if it is not correct). Always assume that the graph $G(V, E)$ is un-directed and connected. Do not assume the edge weights are distinct unless specifically stated. $M$ denote the minimum spanning tree of $G$ and $P$ denote the shortest path between the pair of vertices $v_1$ and $v_2$.
   (a) If each edge weight $w_e$ is replaced by ${w_e}^2$, then $M$ remains the Minimum Spanning Tree of the modified graph.
   (b) If each edge weight $w_e$ is replaced by ${w_e}^2$, then $P$ remains the Shortest Path between $v_1$ and $v_2$ in the modified graph.
   (c) If each edge weight $w_e$ is replaced by $w_e + 1$, then $M$ remains the Minimum Spanning Tree of the modified graph.
   (d) If each edge weight $w_e$ is replaced by $w_e + 1$, then $P$ remains the Shortest Path between $v_1$ and $v_2$ in the modified graph.

(e) If each edge weight $w_e$ is replaced by $w_e * 2$, then $M$ remains the Minimum Spanning Tree of the modified graph.

(f) If each edge weight $w_e$ is replaced by $w_e * 2$, then $P$ remains the Shortest Path between $v_1$ and $v_2$ in the modified graph.

(g) If each edge weight $w_e$ is replaced by ${w_e}^{0.5}$, then $M$ remains the Minimum Spanning Tree of the modified graph.

(h) If each edge weight $w_e$ is replaced by ${w_e}^{0.5}$, then $P$ remains the Shortest Path between $v_1$ and $v_2$ in the modified graph.

(i) If each edge weight $w_e$ is replaced by $\frac{1}{w_e}$, then $M$ remains the Minimum Spanning Tree of the modified graph.

(j) If each edge weight $w_e$ is replaced by $\frac{1}{w_e}$, then $P$ remains the Shortest Path between $v_1$ and $v_2$ in the modified graph.

6. Let $G(V, E)$ be simple undirected graph. If all the edge weights are distinct, then prove that it has a unique minimum spanning tree.

7. Consider a job scheduling problem where you are given a set of jobs $J = \{j_1, j_2, \ldots, j_n\}$. Each job $j_k$ has a deadline $d_k$ associated, and if we can schedule and finish the job before it's deadline, we get the profit $p_k$. Assuming each job is of unit length and only one job can be scheduled at a time, we need to find out a schedule that will earn maximum profit.
For example, consider the following jobs

| Jobs | $j_1$ | $j_2$ | $j_3$ | $j_4$ | $j_5$ | $j_6$ | $j_7$ |
|---|---|---|---|---|---|---|---|
| Deadlines | 2 | 3 | 1 | 2 | 4 | 3 | 2 |
| Profits | 50 | 25 | 15 | 20 | 30 | 30 | 40 |

The optimal solution scheduling in order $j_1, j_7, j_6, j_5$ to earn profit 150.

(a) Consider a greedy algorithm that schedules the jobs as follows. The jobs are sorted based on deadline (ties to be broken arbitrarily). The job with minimum deadline is scheduled if the current time is less than it's deadline else discarded. The process is repeated till all jobs are either scheduled or discarded.
Prove that the above algorithm fails to generate an optimal solution.

(b) Consider a greedy algorithm that schedules the jobs as follows. The jobs are sorted (in decreasing order) based on profit (ties to be broken arbitrarily). The job with maximum profit is scheduled if the current time is less than it's deadline else discarded. The process is repeated till all jobs are either scheduled or discarded.
Prove that the above algorithm fails to generate an optimal solution.