CSCI-GA.1170-001/002 Fundamental Algorithms November 24, 2015

Solutions to Problem 1 of Homework 9 (10 +(6) points)

Name: GOWTHAM GOLI (N17656180) *Due: Tuesday, November 24*

Consider the following greedy algorithm for the ACTIVITY-SELECTION problem. Select the activity a_i with the shortest duration $d_i = f_i - s_i$. Commit to scheduling a_i. Let S'_i consist of all activities a_j which do not overlap with a_i: namely, either $f_j \leq s_i$ or $f_i \leq s_j$. Recursively solve ACTIVITY-SELECTION on S'_i, scheduling the resulting activities together with a_i.

(a) (2 pts) Give a simple example of the input where the proposed greedy algorithm fails to compute the correct optimal solution.

Solution:

Consider the following activities, a_1, a_2 and a_3 such that $(s_1, f_1) = (1, 5), (s_2, f_2) = (4, 7)$ and $(s_3, f_3) = (6, 9)$. By using the above greedy algorithm, the activity selected will be a_2 i.e the activity with the shortest duration and a_1, a_3 will be excluded as they overlap with a_2. Thus the output of the suggested greedy algorithm will be a_2. But clearly, the optimal solution is to select the activities a_1 and a_3. Therefore the proposed greedy algorithm fails to compute the optimal solution $\square$

(b) (4 pts) Consider the following attempt to nevertheless justify the correctness of this algorithm using the "Greedy Always Ahead" method. Given a solution Z, let $F_i(Z)$ be the sum of the i shortest activities scheduled by Z, and ∞ if $|Z| < i$. Try to tell exactly where the claim "For any i and Z, $F_i(Z) \geq F_i(greedy)$" fails. Is is base of induction? Or, if in the inductive step, for what smallest i does the transition from i to $i + 1$ fails? Justify your answer.

Solution:

Base Case

For $i = 1$, The greedy strategy chooses the activity of the shortest duration. Hence $F_1(greedy)$ will be lesser than F_1 of any other feasible solution $\implies F_1(greedy) \leq F_1(Z)$. Therefore the base case is true

Induction Hypothesis

Assume that $F_i(greedy) \leq F_i(Z) \;\; \forall i \leq r$

Inductive Step

Let the activities that the greedy strategy selects are $i_1, \ldots, i_r, i_{r+1}$ and the activities that Z selects are $j_1, \ldots, j_r, j_{r+1}$ We need to prove that $F_{r+1}(greedy) \leq F_{r+1}(Z)$

Let us assume that $F_{r+1}(greedy) > F_{r+1}(Z)$. From the induction hypothesis, we know that $F_r(greedy) \leq F_r(Z)$. Therefore for $F_{r+1}(greedy) > F_{r+1}(Z)$ to happen it must be the case that $d_{i_{r+1}} > d_{j_{r+1}}$. Also $d_{i_{r+1}} > d_{i_r}$. However we cannot make a comparison between d_{i_r} and $d_{j_{r+1}}$.

If $d_{i_r} < d_{j_{r+1}} \implies d_{i_r} < d_{j_{r+1}} < d_{i_{r+1}}$ then the greedy strategy would have selected j_{r+1} instead of i_{r+1} thus contradicting our assumption. Therefore, $F_{r+1}(greedy) \leq F_{r+1}(Z)$.

Else If $d_{j_{r+1}} < d_{i_r} \implies d_{j_{r+1}} < d_{i_r} < d_{i_{r+1}}$ then j_{r+1} would have already been selected by the greedy strategy i.e $j_{r+1} \in \{i_1, \dots i_r\}$ thus our assumption might indeed be true. Therefore, $F_{r+1}(greedy) > F_{r+1}(Z)$

Also if greedy algorithm schedules k activities and Z schedules m activities where $m > k$ then if the given condition were true it would fail from k to $k+1$ as $F_{k+1}(greedy) = \infty$

□

(c) (4 pts) Consider the following alternative attempt to justify the correctness of this algorithm using the "Local Swap" method. Given a solution Z, try to argue that it is always safe to substitute one activity in Z by the first activity of the greedy algorithm, and then argue that the resulting recursive subproblems are the same for the "perturbed" opt and the greedy. Where does your argument run into problem? Is it the swap, or the recursive subproblem part?

Solution:

Let a be the activity of shortest duration in the given set of activities and S be the set of all the activities that overlap with a.

Let the activities scheduled by Z be $OPT = \{a_{j_1}, \dots, a_{j_k}, \dots, a_{j_m}\}$ and OPT^* be the set of activities obtained after substituting some activity in OPT by a. Now we have two cases

If OPT already contains a in it, then we could simply replace the existing a with a. In this case, $OPT^* = OPT$. But if we substitute some a_k by a, we have two a's in OPT^*. So exclude one of the duplicates then $|OPT^*| = |OPT| - 1$. Therefore, this is not an optimal solution anymore.

If OPT doesn't contain a in it and it has one or more non-overlapping activities from the set S. Now replace some activity a_k in OPT with a. Now OPT^* has some overlapping activities in it. So eliminate all those activites that overlap with a from OPT^*. So $|OPT^*| < |OPT|$. Therefore, this is not an optimal solution anymore.

Therefore the argument runs into a problem when we swap some activity in Z with the first activity of the greedy algorithm □

(d) (**Extra credit**) (6 pts) Let t_{opt} be the size of the optimum solution and t_{bogus} be the size returned by the bogus greedy algorithm discussed. Argue that $t_{bogus} \geq t_{opt}/2$. To do this, argue that the first activity a scheduled by the bogus greedy algorithm overlaps *at least* one and *at most* two activities scheduled by the correct optimum algorithm.

In case it overlaps two activities a_1 and a_2, argue that the recursive subproblem resulting from scheduling a has the optimum value at least as large as the one resulting from excluding a_1 and a_2 from opt. (Then use induction on the size of t_{opt}.)

In the the case of only one activity a_1, argue that you can find at most one more activity a_2 in opt such that the recursive subproblem resulting from scheduling a has the optimum value at least as large as the one resulting from excluding a_1 and a_2 from opt. (Then use induction on the size of t_{opt}.) How to you find a_2 if you need it? (This is tricky.)

Solution:

□