CSE 450 Assignment 4

November 22, 2022

Submission Instructions: Deadline is **11:59pm on 11/19/2022 - (Nov 19 2022)**. Late submissions will be penalized, therefore please ensure that you submit (file upload is completed) before the deadline. Additionally, you can download the submitted file to verify if the file was uploaded correctly. Submit your answers electronically, in a single PDF, via *Canvas*. **Please type up the answers and keep in mind that we'll be checking for plagiarism**.
Furthermore, please note that the graders will grade 2 out of the 4 questions randomly. Therefore, if the grader decides to check questions 1 and 4, and you haven't answered question 4, you'll lose points for question 4. Hence, please answer all the questions.

1. In the algorithm for the Activity Selection Problem discussed in class, we order the intervals in increasing order of their finish times and consider them in this order as we schedule non-conflicting intervals. Consider the following variations of the Activity Selection greedy algorithm: **[25]**

   - Sort the intervals in *decreasing* order of their start times and consider them in this order in the algorithm, as we select non-overlapping intervals to include in an optimal solution to the problem.
   - Sort the intervals in *decreasing* order of their lengths and consider them in this order in the algorithm, as we select non-overlapping intervals to include in an optimal solution to the problem.
   - Sort the intervals in *increasing* order of the reciprocal values of their lengths and consider them in this order in the algorithm, as we select non-overlapping intervals to include in an optimal solution to the problem.

   For each of the variations, identify if the strategy will ensure finding the optimal solution for ASP (that is maximum size of mutually compatible activities). If you think that the answer is yes, then please provide your arguments. If your answer is no, then please provide a counter-example.

   ANSWER 1:

   ANSWER 1(a) This strategy WILL guarantee the optimal solution.

   Let us consider a sequence $I =< I_{s_1}, I_{s_2}, ..., I_{s_n} >$ of intervals as an input to the algorithm, where intervals in the sequence are arranged in decreasing order of their start times s_i ($s_i \geq s_{i+1}$). The first interval to be scheduled is I_{s_1}, because it has the highest starting time. Now, if the subsequent interval $I_{s_{i+1}}$, with start-time $s_{i+1} < s_i$, has a finish time $t_{i+1} > s_i$, then it will be discarded (i.e., overlaps are discarded), otherwise will be scheduled. Since this algorithm removes any chances of conflicts, it will guarantee scheduling of non-conflicting intervals when it terminates. Selecting an interval from I that minimizes the difference between its start time with that of the last scheduled interval increases the room for accommodating the remaining intervals in I, and consequently, will maximize the number of intervals scheduled. This strategy will work exactly for the same reason as when the intervals sorted by increasing finish time works.

   ANSWER 1(b) This strategy WILL NOT guarantee the optimal solution.

   This can be disproved by considering the following counter-example:

   Consider three intervals:

   A : [start time: 3, finish time: 15]

   B : [start time: 6, finish time: 8]

   C : [start time: 10, finish time: 13]

   If this strategy is used, interval A will be chosen first and scheduled, leaving no room for scheduling B and C. However, the optimal answer to this problem instance will be scheduling B and C, instead of scheduling A.

   ANSWER 1(c) This strategy WILL NOT guarantee the optimal solution.

   This can be disproved by considering the same counter-example:

Consider three intervals:

A : [start time: 3, finish time: 15]

B : [start time: 6, finish time: 8]

C : [start time: 10, finish time: 13]

If we are to arrange the intervals with respect to the increasing values of the reciprocal of their lengths, then:

A : $\frac{1}{15-3} = \frac{1}{12}$

B : $\frac{1}{8-6} = \frac{1}{2}$

C : $\frac{1}{13-10} = \frac{1}{3}$

Since, $\frac{1}{12} < \frac{1}{3} < \frac{1}{2}$

Therefore, A will be scheduled first, leaving no room for scheduling B and C. However, as seen in the previous example, the optimal answer to this problem instance will be scheduling B and C, instead of scheduling A.

If this strategy is used, interval A will be chosen first and scheduled, leaving no room for scheduling B and C. However, an optimal answer to this problem instance will be scheduling B and C, instead of scheduling A.

2. Consider a single server that has n jobs to be done. The time taken to run each job Ji is ti which is known in advance. The jobs should be finished in the order that minimizes the average waiting time for the jobs. **[25]**

   - Give a greedy algorithm that will find this order and prove its correctness.
   - Can there be multiple schedules which give minimum average waiting time if the time taken for each of the job is distinct, that is no two jobs have same running time? Explain.

   Solution: For the greedy algorithm and proof of correctness refer to the "Minimizing time in the system" algorithm and proof in the lecture slides. Considering jobs in the non-decreasing order of running time will give an optimal solution with minimum average waiting time. When all the running times are distinct there is only one order in which the running times of jobs are in non-decreasing order. So if all the running times are distinct there can only be one optimal schedule.

3. Let $G = (V, E)$ be an undirected graph. A node cover of G is a subset U of the vertex set V such that every edge in E is incident to at least one vertex in U. A minimum node cover is one with the fewest number of vertices. Consider the following greedy algorithm for this problem:

   (a) Procedure COVER(V, E)
   (b) $U \leftarrow \phi$
   (c) loop
   (d) Let $v \in V$ be a vertex of maximum degree
   (e) $U \leftarrow U \cup \{v\}; V \leftarrow V - \{v\}$
   (f) $E \leftarrow E - \{(u, w)\}$ such that $u = v$ or $w = v$
   (g) until $E = \phi$ go to (c)
   (h) return (U)
   (i) end COVER

   Does this algorithm always generate a minimum node cover? If yes, provide some arguments as to why you think so. Else, then provide a counter example where this algorithm fails. **[25]**

   Solution: This algorithm will NOT always generate a minimum node cover. Consider the following example: A tree with the root node denoted as 1, it's children are 2, 3 and 4. 2 has 5 as its child, 3 has 6 as its child and 4 has 7 as its child. The algorithm above will determine that node 1 is the vertex of maximum degree (= 3) and add it to the solution set. Next, note that all the remaining nodes have maximum degree = 1 and there are three edges left in the tree. Therefore, three nodes must be added to the solution set to cover all the nodes in the original tree. Thus, the cardinality of the solution set following this algorithm will be 4. However, the optimal cardinality is 3, when we just select nodes (2, 3, 4). Therefore, this algorithm will not always produce a minimum node cover.

4. For the network shown in Figure **??** above, compute the maximum flow from Vancouver to Winnipeg using the Ford-Fulkerson algorithm discussed in class. **Show all your work**. **[25]**

Solution: The maximum flow in the given network is 23. The minimum cut $C(S : \bar{S})$ is across $S = \{S, v_1, v_2, v_4\}$ and $\bar{S} = \{v_3, T\}$.
Need to be done using ford-fulkerson algortihm discussed in class by picking augmenting paths with forward or backward labeling.