Question 11 Solution

COMP3121/9101 21T3 Final Exam

This document gives a model solution to question 11 of the final exam. Note that alternative solutions may exist.

1. You are given n intervals on an axis. The ith interval $[l_i, r_i)$ has integer endpoints $l_i < r_i$ and has a score of s_i. Your task is to select a set of disjoint intervals with maximum total score. Note that if intervals i and j satisfy $r_i = l_j$ then they are still disjoint.

   Design an algorithm which solves this problem and runs in $O(n^2)$ time.

   You must provide reasoning to justify the correctness and time complexity of your algorithm.

   The input consis ... rs $l_1, r_1, \ldots, l_n, r_n$ and n positive real nu

   The output is the se ... .

   For example, if $n = 4$ and the intervals are:

$l_1 = 0$	$r_1 = 3$	
$l_2 = 1$	$r_2 = 3$	$s_2 = 1$
$l_3 = 2$	$r_3 = 4$	$s_3 = 4$
$l_4 = 3$	$r_4 = 5$	$s_4 = 3$

   then you should select only the first and fourth intervals, for a maximum total score of 5. Note that interval 3 is not disjoint with any other interval.

   Sort the intervals by increasing order of endpoint r_i and relabel accordingly. Henceforth we assume $r_1 \leq \ldots \leq r_n$.

   We then proceed by dynamic programming.

   Subproblems: for $0 \leq i \leq n$, let $P(i)$ be the problem of determining $\text{opt}(i)$, the maximum total score of a set of disjoint intervals where the last chosen interval is the ith, and $m(i)$, the second largest interval index in one such set. If the set consists of only one interval, then $m(i)$ will be zero.

   Recurrence: for $1 \leq i \leq n$,

   $$m(i) = \operatorname*{argmax}_{j : r_j \leq l_i} \text{opt}(j)$$

   and

   $$\text{opt}(i) = s_i + \text{opt}(m(i)).$$

The solution for i must include interval i, so we extend the best solution with last chosen interval j finishing at or before the start of interval i.

Base case: $\text{opt}(0) = 0$ and $m(0)$ is undefined.

Order of computation: subproblem $P(i)$ depends only on earlier subproblems ($P(j)$, where $j < i$), so we can solve the subproblems in increasing order of i.

Final answer: The maximum total score is

$$\max_{1 \le i \le n} \text{opt}(i).$$

To recover the set which yields this score, we let

$$i^* = \operatorname*{argmax}_{1 \le i \le n} \text{opt}(i),$$

and backtrack through the m array to obtain the set $\{i^*, m(i^*), m(m(i^*)), \ldots\}$.

Time complexity: There are $O(n)$ subproblems, each solved in $O(n)$, and constructing the final answer also takes $O(n)$. Thus the overall time complexity is $O(n^2)$.