CS207 Design and Analysis of Algorithms

Sajith Gopalan

Indian Institute of Technology Guwahati

sajith@iitg.ac.in

February 18, 2022

Greedy Method

Greedy Method

- Works for some optimization problems, not all
- A greedy choice is one that looks best at the moment
- Steps:
  - Formulate a solution in which we make a *greedy* choice and are left with one subproblem.
  - Prove that there is always an optimal solution reachable thorugh the greedy choice; that is, the greedy choice is safe
  - Devise a way to combine an optimal solution to the subproblem with the greedy choice to obtain an optimal solution to the original problem

The Interval Selection Problem

- Consider a set of intervals $S = \{a_1, \ldots, a_n\}$ where $a_i = (s_i, f_i)$
- It is required to find a maximum cardinality subset of non-overlapping intervals (MCSNI) A of S
- A dynamic programming solution can be built as follows:
- Say, S_{ij} denote the set of activities that start after activity a_i finishes and finish before activity a_j starts.
- Let $c[i, j]$ denote the cardinality of the MCSNI A_{ij} of S_{ij}

The Interval Selection Problem

- Then

$$c[i,j] = \left\{ \begin{array}{ll} 0 & \text{if } S_{ij} = \emptyset \\ \max_{a_k \in S_{ij}} \{1 + c[i,k] + c[k,j]\} & \text{otherwise} \end{array} \right\}$$

- So, $c[0, n+1]$ is the cardinality of the MCSNI A of S
- The set itself can be constructed by riding piggyback on the the above construction.
- The resultant algorithm runs in $O(n^2)$ time.

The Interval Selection Problem

- But we can do better
- Being greedy helps!
- If S is sorted on finish times in acending order, a_1 is the first interval to finish
- There is an MCSNI A of S so that $a_1 \in A$
- Proof: Let A be an MCSNI of S. If $a_1 \in A$, then we are done. So assume $a_1 \notin A$. Let a_i be the member of A that has the least finish time. Every other member of A starts after a_i finishes. Intervals a_1 and a_i overlap; otherwise a_1 could be added to A to get a larger cadinality subset of S, where no two members overlap, which would be a contradiction as A is an MCSNI. Replace a_i with a_1 in A, and we get another MCSNI of S. Hence the claim.

The Interval Selection Problem

- The above finding suggests an algorithm:
- Pick the interval with the least finish time: Greedy Choice
- Remove all intervals overlapping with it
- Recurse with the remaining set

The Interval Selection Problem: Algorithm

Algorithm 1 IntervalSelection(s, f)

Sort the intervals on finish time. Let $a_1, \ldots, a_n$ be the sorted order.
$A = \{a_1\}$; $last = 1$;
for $i = 2$ to n **do**
    if $s[i] \geq f[last]$ **then**
        $A = A \cup \{a_i\}$
        $last = i$;
    end if
end for
return A

Analysis

- The sorting of the intervals takes $O(n \log n)$ time
- The rest of the algorithm takes $O(n)$ time
- Thus, the time complexity id $O(n \log n)$
- The greedy choice leaves only one subproblem to consider: the set of intervals that do not overlap a_1
- The Dynamic Programming solution looks at many subproblems. That turns out to be unnecessary in this problem.