San Skulrattanakulchai
Gustavus Adolphus College Oct 24, 2014

Activity Selection

A simple way to solve a problem is the *greedy strategy*. This works as follows. To find an optimal solution, we construct the solution by making decisions to choose each item to add to the solution optimally, and never reversing decisions already made.

Example

Coin Changing Problem: We want to make change for c cents using coins of denominations 25¢, 10¢, 5¢, and 1¢, and using the fewest possible number of coins.
A greedy algorithm for the problem runs like this:
use as many 25¢ as possible, then
use as many 10¢ as possible, then
use as many 5¢ as possible, then finally
use as many 1¢ as possible
This algorithm works correctly for these denominations. However, for the denominations 25¢, 10¢, and 1¢, the greedy strategy fails. For example, to make change for 30 cents, the greedy algorithm uses 6 coins (25 + 1 + 1 + 1 + 1 + 1). However, we can use 3 coins (10 + 10 + 10) instead.

Exercise

Show that the greedy strategy also fails for the denominations 25¢, 20¢, 10¢, 5¢, and 1¢.

Problem

Given n classes, each having a start time and finish time, we would like to schedule as many classes as possible in one lecture hall. No two classes can use the hall at the same time.
Example Given classes [0, 4), [3, 5), [1, 6), [4, 7)
An optimal schedule has 2 classes: [0, 4) and [4, 7).

Algorithm

1. Sort the classes into nondecreasing order using their finish times as key.

Now we have classes $j = 1, \ldots, n$ with start times $S[j]$, and finish times $F[j]$, with $F[1] \leq F[2] \leq \cdots \leq F[n-1] \leq F[n]$.

2. Find the optimum schedule as follows.

> $\mathcal{S} \leftarrow \emptyset$
> /* we maintain variable f equal to the finish time of the last class scheduled */
> $f \leftarrow -\infty$
> **for** $j \leftarrow 1$ to n **do** {
>   **if** $S[j] > f$ **then** {
> $\mathcal{S} \leftarrow \mathcal{S} \cup \{j\}$
> $f \leftarrow F[j]$
>   }
> }
> **return** $\mathcal{S}$ as the desired schedule

Correctness

Correctness depends on the following theorem.

Theorem. *Some optimal schedule contains the class with the earliest finish time.*

Proof. Let $\mathcal{S}$ be an optimal schedule and let C be a class with an earliest finish time. If the first class scheduled by $\mathcal{S}$ is C, we are done. If not, let C' be the first class scheduled by $\mathcal{S}$. Let $\mathcal{S}' = (\mathcal{S} - \{C'\}) \cup C$. Then $\mathcal{S}'$ is a valid schedule of the same size as $\mathcal{S}$ and thus optimal. Moreover, it starts with C as desired. □

Exercise Which of the following greedy strategy (if any) also gives an optimal schedule: earliest start-time first, latest start-time first, latest finish-time first, shortest class-time first, longest class-time first?

Running Time

Step 1 uses time $O(n \log n)$ using heap sort, for example.

Step 2 uses time $O(n)$ since we process each compound in time $O(1)$.

Remark

This problem is the Maximum Independent Set Problem on interval graphs. Even though the Maximum Independent Set Problem on general graphs is NP-complete, this handout shows it is solvable in polynomial time when restricted to interval graphs.