Greedy 1: Activity Selection

Yufei Tao

Department of Computer Science and Engineering
Chinese University of Hong Kong

In this lecture, we will commence our discussion of **greedy** algorithms, which enforce a simple strategy: make the **locally optimal** decision at each step. Although this strategy does not always guarantee finding a **globally optimal** solution, sometimes it does. The nontrivial part is to prove (or disprove) the global optimality.

Activity Selection

Input: A set S of n intervals of the form $[s, f]$ where s and f are integers.
Output: A subset T of disjoint intervals in S with the largest size $|T|$.

> **Remark:** You can think of $[s, f]$ as the duration of an activity, and consider the problem as picking the largest number of activities that do not have time conflicts.

Activity Selection

Example: Suppose

$$S = \{[1,9],[3,7],[6,20],[12,19],[15,17],[18,22],[21,24]\}.$$

$T = \{[3,7],[15,17],[18,22]\}$ is an optimal solution, and so is $T = \{[1,9],[12,19],[21,24]\}$.

Activity Selection

Algorithm

Repeat until S becomes empty:

- Add to T the interval $\mathcal{I} \in S$ with the smallest finish time.
- Remove from S all the intervals intersecting $\mathcal{I}$ (including $\mathcal{I}$ itself)

Activity Selection

Example: Suppose $S = \{[1, 9], [3, 7], [6, 20], [12, 19], [15, 17], [18, 22], [21, 24]\}$.

Sort the intervals in S by finish time: $S = \{[3, 7], [1, 9], [15, 17], [12, 19], [6, 20], [18, 22], [21, 24]\}$.

We first add $[3, 7]$ to T, after which intervals $[3, 7]$, $[1, 9]$ and $[6, 20]$ are removed. Now S becomes $\{[15, 17], [12, 19], [18, 22], [21, 24]\}$. The next interval added to T is $[15, 17]$, which shrinks S further to $\{[18, 22], [21, 24]\}$. After $[18, 22]$ is added to T, S becomes empty and the algorithm terminates.

Activity Selection

Next, we will prove that the algorithm returns an optimal solution. Let us start with a crucial claim.

Claim: Let $\mathcal{I} = [s, f]$ be the interval in S with the smallest finish time. There must be an optimal solution that contains $\mathcal{I}$.

Proof: Let T^* be an arbitrary optimal solution that does not contain $\mathcal{I}$. We will turn T^* into another optimal solution T containing $\mathcal{I}$.

Let $\mathcal{I}' = [s', f']$ be the interval in T^* with the **smallest** finish time. We construct T as follows: add all the intervals in T^* to T **except** $\mathcal{I}'$, and finally add $\mathcal{I}$ to T.

We will prove that all the intervals in T are disjoint. This indicates that T is also an optimal solution, and hence, will complete the proof.

Activity Selection

It suffices to prove that $\mathcal{I}$ cannot intersect with any other interval in T.

Consider any interval $\mathcal{J} = [a, b]$ in T. By definition of $\mathcal{I}'$, we must have $f' \leq b$. Combining this and the fact that $\mathcal{J}$ is disjoint with $\mathcal{I}'$, we assert that $f' < a$. On the other hand, by definition of $\mathcal{I}$, it must hold that $f \leq f'$. It thus follows that $f < a$ and, hence, $\mathcal{I}$ and $\mathcal{J}$ are disjoint.

□

Activity Selection

Think 1: How to utilize the claim to prove that our algorithm is optimal?

Think 2: How to implement the algorithm in $O(n \log n)$ time?