COMPSCI 311: Introduction to Algorithms

Lecture 7: Greedy Algorithms

Dan Sheldon

University of Massachusetts Amherst

Greedy Algorithms

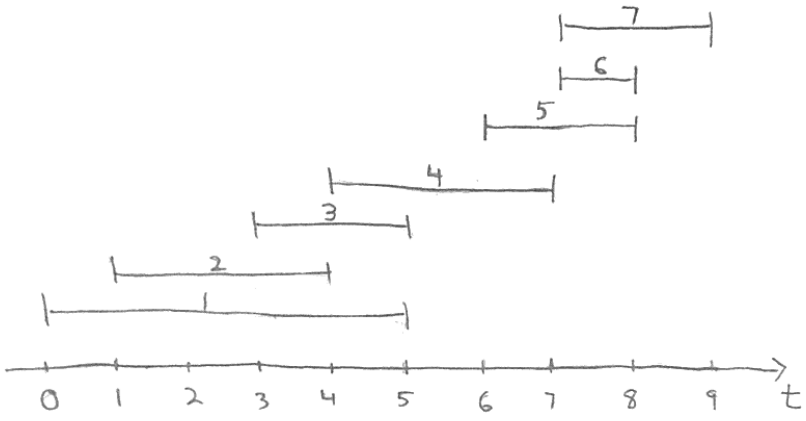
We are moving on to our study of algorithm design techniques:

- **Greedy**
- Divide-and-conquer
- Dynamic programming
- Network flow

Let's jump right in, then characterize later what is means to be "greedy".

Interval Scheduling

- In the 80s, you could only watch a given TV show at the time it was broadcast. You want to watch the highest number of shows. Which subset of shows do you pick?



Formalizing Interval Scheduling

Let's formalize the problem

- Shows $1, 2, \ldots, n$ (more generally: *requests* to be fulfilled with a given resource)
- s_j: start time of show j
- f_j, also written $f(j)$: finish time of show j

- Shows i and j are compatible if they don't overlap.
- Set A of shows is compatible if all pairs in A are compatible.
- Set A of shows is optimal if it is compatible and no other compatible set is larger.

Goal: find optimal set of shows

Greedy Algorithms

- Main idea in greedy algorithms is to make one choice at a time in a "greedy" fashion. (Choose the thing that looks best, never look back...)
- We will sort shows in some "natural order" and choose shows one by one if they're compatible with the shows already chosen.
  Concretely:

$R \leftarrow$ set of all shows **sorted by some property**
$A \leftarrow \{\}$ ▷ selected shows
while R is not empty **do**
    take first show i from R
    add i to A
    delete i and all overlapping shows from R

Clicker

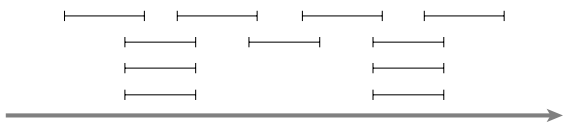
$R \leftarrow$ set of all shows **sorted by some property**
$A \leftarrow \{\}$ ▷ selected shows
while R is not empty **do**
    take first show i from R
    add i to A
    delete i and all overlapping shows from R

Suppose an algorithm includes a step that sorts an n items. Then its running time is:

A. $O(n \log n)$
B. $\Omega(n \log n)$
C. $\Theta(n \log n)$
D. None of the above

What's a "natural order" ?

- *Start Time*: Consider shows in ascending order of s_j?
  Not optimal in running example.
- *Shortest Time*: Consider shows in ascending order of $f_j - s_j$?
  Not optimal in running example.
- *Fewest Conflicts*: Let c_j be number of shows which overlap with show j. Consider shows in ascending order of c_j.
  Optimal in running example. But not this one:



(c)

- *Finish Time*: Consider shows in ascending order of f_j.
  We'll show that this is always optimal!

Analysis

Let A be the set of shows returned by the algorithm when shows are sorted by finish time. What do we need to prove?

- A is compatible (obvious property of algorithm)
- A is optimal

We will prove A is optimal by a "greedy stays ahead" argument

Ordering by Finish Time is Optimal: "Greedy Stays Ahead"

- Let $A = i_1, \ldots, i_k$ be the intervals selected by the greedy algorithm
- Let $O = j_1, \ldots, j_m$ be the intervals of some optimal solution O
- Assume both are sorted by finish time

```
A: |--i1--||---i2---| ... |---ik---|
O: |---j1---||---j2---|      ...      |----jm----|
```

- Could it be the case that $m > k$?
- Observation: $f(i_1) \leq f(j_1)$. The first show in A finishes no later than the first show in O.
- **Claim** ("greedy stays ahead"): $f(i_r) \leq f(j_r)$ for all $r = 1, 2, \ldots$.
  The rth show in A finishes no later than the rth show in O.

"Greedy Stays Ahead"

- **Claim**: $f(i_r) \leq f(j_r)$ for all $r = 1, 2, \ldots$
- **Proof** by induction on r
- **Base case** ($r = 1$): i_r is the first choice of the greedy algorithm, which has the earliest overall finish time, so $f(i_r) \leq f(j_r)$

Induction Step

- Assume inductively that $f(i_{r-1}) \leq f(j_{r-1})$ $(r \geq 2)$

```
A: |--i1--| ... |---i(r-1)---|
O: |---j1---| ... |---j(r-1)---||----jr-----|
```

- j_r is compatible with j_{r-1}, so $s(j_r) \geq f(j_{r-1})$
- $f(j_{r-1}) \geq f(i_{r-1})$ by inductive hypothesis
- Thus, $s(j_r) \geq f(i_{r-1})$ and interval j_r is in the set of available intervals when trying to select i_r
- Since we greedily select the earliest finish time, $f(i_r) \leq f(j_r)$, completing the inductive step

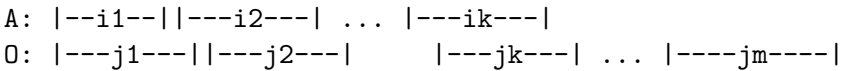
Clicker

```
A: |--i1--||---i2---| ... |---ik---|
O: |---j1---||---j2---|     ...      |----jm----|
```

Recall that k is the number of intervals in the greedy solution and m is the number of intervals in an optimal solution. What have we just proven?

A. $f(i_r) \leq f(j_r)$ for $r = 1, 2, \ldots, m$
B. $f(i_r) \leq f(j_r)$ for $r = 1, 2, \ldots, k$
C. The greedy algorithm is optimal.
D. None of the above.

Optimality



Can it be the case that $k < m$?

No. Because "greedy stays ahead", intervals j_{k+1} through j_m would be compatible with the greedy solution, and the greedy algorithm would not terminate until adding them.

Running Time?

$R \leftarrow$ set of all shows **sorted by finishing time**
$A \leftarrow \{\}$
while R is not empty **do**
    take first show i from R
    add i to A
    delete i and all overlapping shows from R ▷ $O(n)$?

Can we make loop better than n^2?

Running Time?

$R \leftarrow$ set of all shows **sorted by finishing time**
$A \leftarrow \{\}$, end $= 0$ ▷ most recent end time
for show i from 1 to n **do**
  if $s_i \geq$ end **then**
    add i to A; end $= f_i$ ▷ $O(1)$

$\Theta(n \log n)$ — dominated by sort

Algorithm Design—Greedy

Greedy: make a single "greedy" choice at a time, don't look back.

Learning goals:

	Greedy
Formulate problem	
Design algorithm	
Prove correctness	✓
Analyze running time	
Specific algorithms	Dijkstra, MST

Focus is on proof techniques. Next: another proof technique.