LECTURE 7

GREEDY ALGORITHMS

ACTIVITY SELECTION
Greedy Algorithms: Algs that iteratively make "short-sighted", locally optimum looking decisions with the hope that they output an optimal solution.

Ex: Coin changing:

Counter example:

<table>
<thead>
<tr>
<th>10 cents</th>
<th>14 cents</th>
<th>11 cents</th>
</tr>
</thead>
<tbody>
<tr>
<td>7 x 1</td>
<td>4 x 1</td>
<td>2 x 7</td>
</tr>
<tr>
<td>7 coins</td>
<td>5 coins</td>
<td>2 coins</td>
</tr>
</tbody>
</table>
### Comparison of Greedy vs D&C

<table>
<thead>
<tr>
<th>Greedy</th>
<th>D&amp;C</th>
</tr>
</thead>
<tbody>
<tr>
<td>easy to design</td>
<td>difficult to design</td>
</tr>
<tr>
<td>easy to analyze</td>
<td>difficult to analyze</td>
</tr>
<tr>
<td>difficult to prove correctness</td>
<td>easy to prove correctness</td>
</tr>
</tbody>
</table>

---

2. Main Proof Techniques

1. Greedy-stays-ahead arguments:
   - argue that the greedy solution \( S_g \) is better than an arbitrary solution \( S \) at each iteration.

2. Exchange Argument:
   - argue that an arbitrary solution \( S \) can be transformed step by step & without getting worse into the greedy solution \( S_g \).
Activity Selection:

Input: A resource & n requests, where each request i has a start time $s(i)$ and a finishing time $f(i)$.

Output: Max # of non-overlapping requests.
Greedy Alg 1: The earliest start time, first
Greedy Alg 2: Shortest greedy algorithm first
Greedy Alg 3: Min-overlapping requests first
Greedy Alg 4: The earliest finishing time first
**Proof of optimality of EFT Greedy alg:**

Let $S_g = \{ g_1, g_2, \ldots, g_k \}$ be the $k$ requests that greedy EFT picks.

**Key Claim:** Earliest time any schedule can fulfill $i$ requests is $f(g_i)$.

**Proof:** By induction on $i$.

**Base Case:** $i = 1$, holds bc $f(g_1)$ is the earliest $f_i$ of all requests.

**Inductive Hypothesis:** Suppose the claim holds for $i = t$. Prove for $i = t + 1$. 


let \( r_{h_1} r_{h_2} \ldots r_{h_t} r_{h_{t+1}} \) be the first test requests executed by an arbitrary solution.

Suppose for contradiction that
\[
\forall t \in \mathbb{N} \quad f(r_{h_{t+1}}) < f(r_{h_t})
\]

Pictorially:
\[
\begin{array}{c}
\text{r_{h_t}} \quad \text{f(r_{h_t})} \\
\downarrow \quad \downarrow \\
\text{r_{h_{t+1}}} \quad \text{f(r_{h_{t+1}})}
\end{array}
\]

But this is a contradiction.

More formally, we have:
\[
f(r_{h_{t+1}}) < f(r_{h_t}) \leq S(r_{h_{t+1}}) \leq f(r_{h_{t+1}}) < f(r_{h_{t+1}})
\]

by \( \text{f} \)

by assumption

for contradiction

\[
\text{the start times are earlier than finish times}
\]
This is a contradiction because the r HuffPost is not overlapping with r HuffPost (or anything greedy eff picked earlier). So greedy eff would have picked r HuffPost our r HuffPost (which is later than r HuffPost).

Claim: Greedy eff is optimal.
Proof: (Similar to the proof of the key claim).

Let $S_g = g_1 \ldots g_k$.

Suppose for contradiction that there is another solution with $k+1$ requests:

$S_r = r_1 \ldots r_{k+1}$.

The pictorially:

But this is a contradiction, i.e., we know by the key claim that $f(g_1) \leq f(r_{k+1})$ so $r_{k+1}$ does not overlap with $E_{g_1}, \ldots, E_{g_k}$ and Greedy would have ended it.
Full Alg: Greedy EFT! 

Input: R: a set of n requests 
Sort R by ft of requests \( \rightarrow \Theta(n \log n) \)

\( S_g = \{ R[0] \} \) 
\( \text{latestFT: } R[0].ft \) 
for \( i = 1, \ldots, n \) 
if \( [R[i], \text{start}] > \text{latestFT} \) 
\( S_g = \text{add} \{ R[i] \} \) 
\( \text{latestFT} = R[i].ft \).

\( \text{Runtime: } \Theta(n \log n) \).