Restaurant

Restaurant orders

\( l(1), \ldots, l(n) \) - how long each order takes to cook

tip = subject $\dagger$ for each

\[ \text{minimize the total waiting time} \]

\[ \text{lengths: 1, 7, 3, 55} \]

\[ 4 \]

\[ 1 \]

\[ 3 \]

\[ 3 \]

\[ 55 \]

\[ 1 \]

\[ 7 \]

\[ 66 \]

\[ 86 \]

\[ 220 \]

\[ 371 \]

Shortest Job First

\[ \text{while jobs left} \]

\[ \text{schedule shortest job} \]

Suppose greedy schedule - \( g \)

\( g(1) \leq g(2) \leq \ldots \leq g(n) \)

Optimal schedule \( \sigma \), on \( g(1), \ldots, g(n) \)

\[ \text{swap } o_k \rightarrow o_{k+1} \]

\[ o_k' = o_{k+1}, \quad o_{k+1}' = o_k \]

\[ w(o_k') = w(o_{k+1}) - l(o_k) \leq w(o_k) = w(o_{k+1}) + l(o_k) \]

\[ w(o_{k+1}) = w(o_k) + l(o_{k+1}) \]

\[ \text{satisfies} \]

\[ \sum_{i=1}^{n} l(i) \]

\[ \text{lengths: 1, 7, 3, 55} \]

\[ 4 \]

\[ 1 \]

\[ 3 \]

\[ 3 \]

\[ 55 \]

\[ 1 \]

\[ 7 \]

\[ 66 \]

\[ 86 \]

\[ 220 \]

\[ 371 \]
\[ g(0, i) \leq g(0, n) \]
\[ \ell(L, 0, i) \leq \ell(L, 0, n) \]

We can produce optimal schedule in greedy order

\[ \begin{array}{cccccccc}
1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\
\hline
1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\
\end{array} \]

\[ 11 - 7 \quad 8 + 3 \]

**Person**

1. Tip 1: Min wait 1 11
2. 2: 5 7 11 7.75
3. 3 7.2 11 3.3 12
4. 4 6.8 11 55.5

order by \[ \ell(i)/\text{tip}(i) \]

**Shortest Job First**

while jobs left

\[ \text{schedule shortest job} \]

\[ \forall i \quad \text{schd}(\text{jobs}) = O(n^2) \]

\[ \text{result} = \{ i \} \]

while \( \text{en}(\text{jobs}) > 0 \)

\[ \forall i \quad \text{result} = \text{append}(\text{result}, \text{en}(\text{jobs})) \]

\[ \text{jobs} = \text{reduce}(\text{jobs}, \text{en}(\text{jobs})) \]

\[ \text{return sorted(\text{jobs})} \]

\[ O(n \log n) \]
Let $f_i$ finishes first. Then $S(o_2) \geq F(o_i)$ and $S(o_2) > F(f_i)$ if $f_2$ finishes first.
Common pattern
- At each step use "greedy heuristic" to select next element
- Exchange argument optimally

Companies: A B C D
Students: a b c d

Internship interviews:
A: a c b d
B: c a d b
C: a c d b
D: a d b c
a: C B D A
b: C D A B
c: D B A C
d: A B C D

Matching: each co to each student
Each round:

1. Find an unmatched company.
2. Offer to best person.
3. Accept if unmatched, otherwise reject.

For any list of preferences and can be found in O(n).

If unstable:

if student prefers X to Y

if student prefers Y to X

Student co-X is unmatched.
reject if matching to better co.