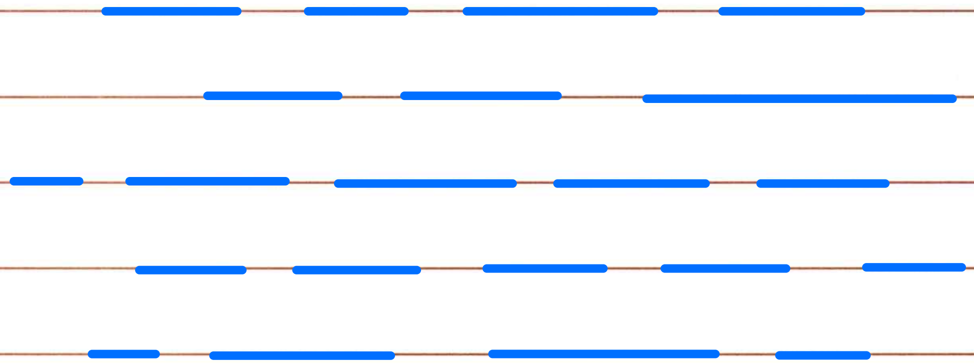
Greedy

Interval Scheduling Problem

Input: A set of requests $\{1..n\}$, where the i^{th} request starts at $s(i)$ and ends at $f(i)$.

Output: A largest compatible subset of these requests.



High Level Solution

```
Initially R is the complete set of
        requests and A is empty
While R is not empty
    Choose a request $i \in R$ that has
        the smallest finish time
    Add request $i$ to A
    Delete all requests from R that
        are not compatible with $i$
Endwhile
Return A
```

Proof of Correctness

1_ Prove that A is a compatible set

2_ Prove that A is an optimal set

Easy to show #1: Since we always delete all overlapping requests before choosing the next request, we can never end up with overlapping requests in A.

#2: Say A is of size k, and suppose there is an optimal solution O.

Label requests in A: $i_1, i_2, \ldots i_k$
" " " O: $j_1, j_2, \ldots, j_m$

We will first prove that for all indices $1 \leq r \leq k$, we have $f(i_r) \leq f(j_r)$

Implementation

Sort requests in order of finish time and label in this order:

$$f(i) \leq f(j) \text{ where } i < j$$

Select requests in order of increasing $f(i)$, always selecting the first request.
Then iterate through the intervals in this order until reaching the first interval j where $s(j) \geq f(i)$ and then pick j.