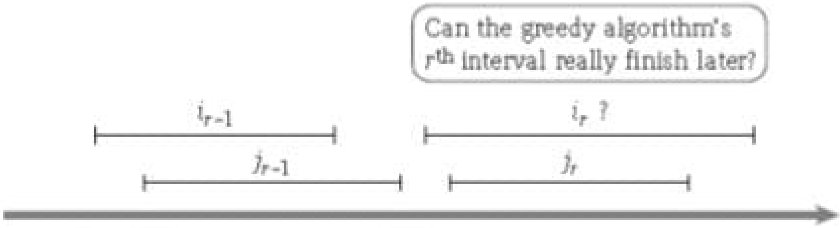
Analysis

1. ***The greedy algorithm returns an optimal set A.***

- In order to prove this we first have to prove that **set A contains a set of compatible requests.**
- Let $i_1, \ldots, i_k$ be the set of requests in A in the order they were added to A. Note that $|A| = k$. Similarly,
- Let the set of requests in optimal set O be denoted by $j_1, \ldots, j_m$. Our goal is to prove that $k = m$.
- Our greedy rule guarantees that $f(i_1) \le f(j_1)$. This is the sense in which we want to show that our greedy rule “stays ahead”—that each of its intervals finishes at least as soon as the corresponding interval in the set O. Thus we now prove that for each $r \ge 1$, the rth accepted request in the algorithm's schedule finishes no later than the rth request in the optimal schedule.
  i.e. *For all indices $r \le k$ we have $f(i_r) \le f(j_r)$.*
- For $r = 1$ the statement is clearly true: the algorithm starts by selecting the request i_1 with minimum finish time.
- Now let $r > 1$. We will assume as our induction hypothesis that the statement is true for $r - 1$, and we will try to prove it for r. As shown in Figure 4.3, the induction hypothesis lets us assume that $f(i_{r-1}) \le f(_{ir-1})$.



- the greedy algorithm always has the option (at worst) of choosing j_r and thus fulfilling the induction step. i.e. j_r is compatible with i_{r-1} and j_{r-1}.
- We know (since O consists of compatible intervals) that $f(j_{r-1}) \le s(j_r)$. Combining this with the induction hypothesis $f(i_{r-1}) \le f(j_{r-1})$, **we get** $f(i_{r-1}) \le s(j_r)$.
- Thus the interval j_r is in the set R of available intervals at the time when the greedy algorithm selects i_r. The greedy algorithm selects the available interval with *smallest* finish time; since interval j_r is one of these available intervals, we have $f(i_r) \le f(j_r)$.
- Thus we have formalized the sense in which the greedy algorithm is remaining ahead of O: for each r, the r^{th} interval it selects finishes at least as soon as the r^{th} interval in O.

We will now prove the statement by contradiction.

- If A is not optimal, then an optimal set O must have more requests, that is, we must have $m > k$.
- With $r = k$, we get that $f(i_k) \le f(j_k)$. Since $m > k$, there is a request j_{k+1} in O.
- This request starts after request j_k ends, and hence after i_k ends.
- So after deleting all requests that are not compatible with requests $i_1, \ldots, i_k$, the set of possible requests R still contains j_{k+1}.
- But the greedy algorithm stops with request i_k, and it is only supposed to stop when R is empty—a contradiction. Hence, the set A returned is optimal