贪心法的例子：
活动选择问题

活动选择问题

输入：$S = \{1, 2, \ldots, n\}$为n 项活动的集合，s_i，f_i 分别为活动 i 的开始和结束时间.

活动 i 与 j 相容 $\Leftrightarrow s_i \geq f_j$ 或 $s_j \geq f_i$.

求：最大的两两相容的活动集 A

输入实例：

i	1	2	3	4	5	6	7	8	9	10
s_i	1	3	2	5	4	5	6	8	8	2
f_i	4	5	6	7	9	9	10	11	12	13

解：$\{1, 4, 8\}$

贪心算法

挑选过程是多步判断，每步依据某种“短视”的策略进行活动选择，选择时注意满足相容性条件.

策略1：开始时间早的优先
排序使 $s_1 \le s_2 \le \ldots \le s_n$，从前向后挑选

策略2：占用时间少的优先
排序使得 $f_1 - s_1 \le f_2 - s_2 \le \ldots \le f_n - s_n$，从前向后挑选

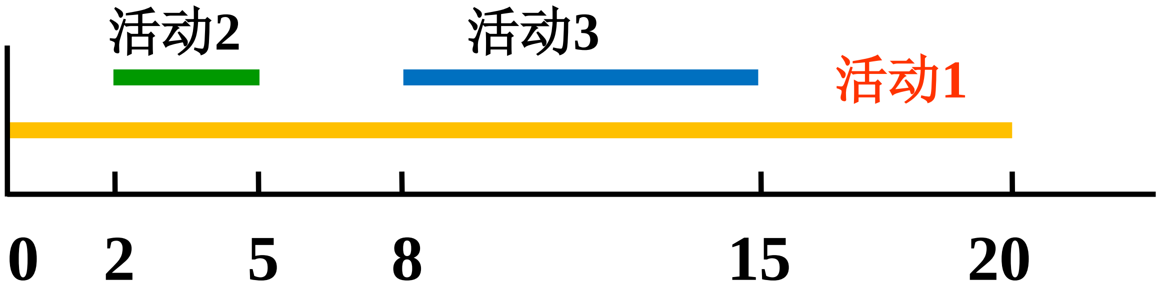
策略3：结束早的优先
排序使 $f_1 \le f_2 \le \ldots \le f_n$，从前向后挑选

策略1的反例

策略1：开始早的优先

反例：$S=\{1,2,3\}$

$s_1=0$，$f_1=20$，$s_2=2$，$f_2=5$，$s_3=8$，$f_3=15$

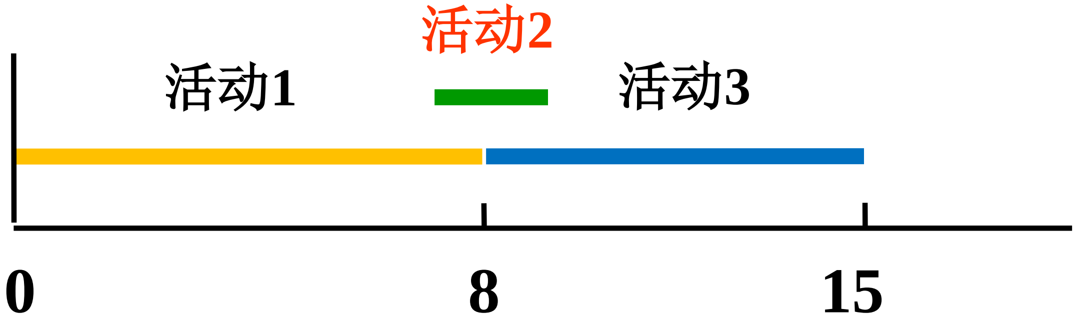


策略2的反例

策略2：占时少的优先

反例： $S = \{ 1, 2, 3 \}$

$s_1=0,\ f_1=8,\ s_2=7,\ f_2=9,\ s_3=8,\ f_3=15$



策略3伪码

算法 Greedy Select

输入：活动集 S，s_i, f_i,

$i = 1, 2, \ldots, n$，$f_1 \le \ldots \le f_n$

输出：$A \subseteq S$，选中的活动子集

1. $n \leftarrow length[S]$
2. $A \leftarrow \{1\}$
3. $j \leftarrow 1$
4. for $i \leftarrow 2$ to n do
5. if $s_i \ge f_j$
6. then $A \leftarrow A \cup \{ i \}$
7. $j \leftarrow i$
8. return A

已选入的
最后标号

判相容

完成时间 $t = \max \{ f_k : k \in A \}$

运行实例

输入：$S = \{ 1, 2, \ldots, 10 \}$

i	1	2	3	4	5	6	7	8	9	10
s_i	1	3	0	5	3	5	6	8	8	2
f_i	4	5	6	7	8	9	10	11	12	13

解：$A = \{1, 4, 8\}$，$t = 11$

时间复杂度

$$O(n\log n) + O(n) = O(n\log n)$$

如何证明该算法对所有的实例都得到正确的解?

贪心算法的特点

设计要素：

(1) 贪心法适用于组合优化问题.

(2) 求解过程是多步判断过程，最终的判断序列对应于问题的最优解.

(3) 依据某种“短视的”贪心选择性质判断，性质好坏决定算法的成败.

(4) 贪心法必须进行正确性证明.

(5) 证明贪心法不正确的技巧：举反例.

贪心法的优势：算法简单，时间和空间复杂性低