六、典型算法

6.1 贪心算法

6.1.1 贪心思想

6.1.2 活动选择案例

6.1.3 装载问题案例

6.1.4 其他案例

贪心算法：总是作出在当前看来最好的选择（局部最优解）。

关键：每一步如何做出局部最优解选择

注意：贪心算法不能对所有问题都得到整体最优解！

可用贪心算法求解问题的2个重要的性质：

贪心选择性质

最优子结构性质

贪心算法性质

可用贪心算法求解问题的2个重要的性质：

贪心选择性质

最优子结构性质

贪心选择性质：所求问题的**整体最优解可以通过一系列局部最优的选择**，即贪心选择来达到。

对于一个具体问题，要确定它是否具有贪心选择性质，必须**证明**每一步所作的贪心选择最终导致问题的整体最优解。

贪心算法性质

贪心选择性质：所求问题的**整体最优解可以通过一系列局部最优的选择**，即贪心选择来达到。

证明过程:

1. 考察问题的一个整体最优解，并证明可修改这个最优解，以**贪心选择开始**
2. 贪心选择后，原问题简化为规模更小的**子问题**
3. 用**数学归纳法**证明，通过每步贪心选择，最终可得到问题的整体最优解

贪心算法性质

可用贪心算法求解问题的2个重要的性质：

贪心选择性质

最优子结构性质

最优子结构性质： 当一个问题的最优解包含其子问题的最优解时，称此问题具有最优子结构性质。

问题的最优子结构性质是该问题可用分治递归算法、动态规划算法或贪心算法求解的关键特征。

六、典型算法
6.1 贪心算法

6.1.1 贪心思想

6.1.2 活动选择案例

6.1.3 装载问题案例

6.1.4 其他案例

设有n个活动的集合S={1, 2, …, n}，每个活动i都有各自的起始时间 s_i和一个结束时间 f_i 且 $s_i < f_i$

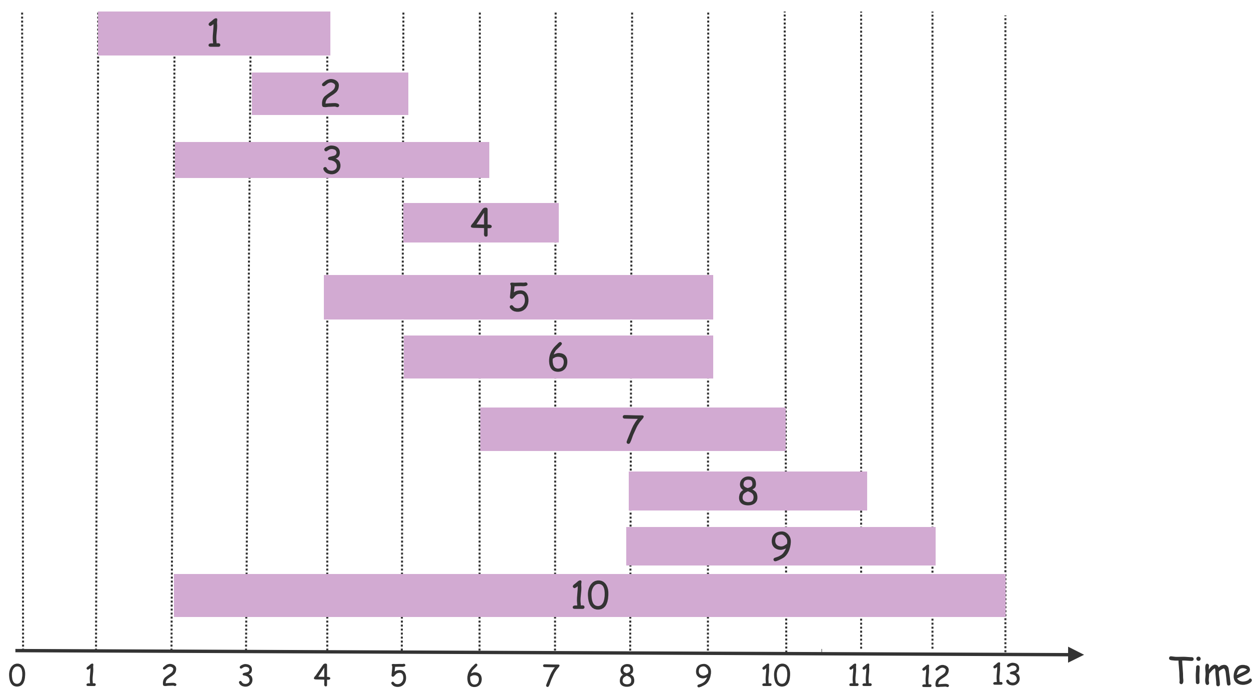
每个活动都要求使用同一资源，如计算资源或演讲会场等，而在同一时间内只有一个活动能使用这一资源

如果选择了活动i，则它在半开时间区间[si, fi)内占用资源，此时间段内无法安排其他活动。

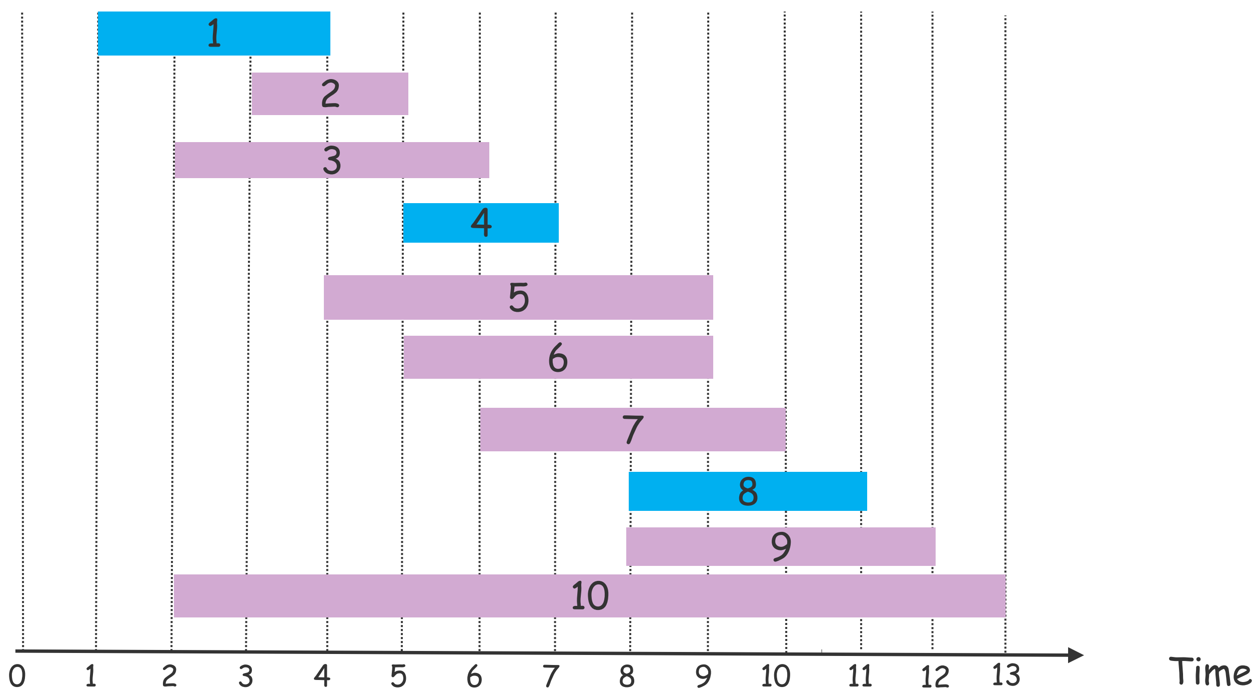
活动 i 和 j 相容：有 $s_i \geq f_j$ 或 $s_j \geq f_j$

希望选择包含尽可能多相容的活动的集合

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



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}

贪心挑选过程是多步判断过程，每步依据某种“贪心”策略进行活动选择

活动选择策略：

1. **开始时间早的优先：**
   依据$s_1 \le s_2 \le \dots \le s_n$ 依次向后挑选出相容的活动

2. **占用时间短的优先：**
   依据$f_1-s_1 \le f_2-s_2 \le \dots \le f_n-s_n$依次向后挑选出相容的活动

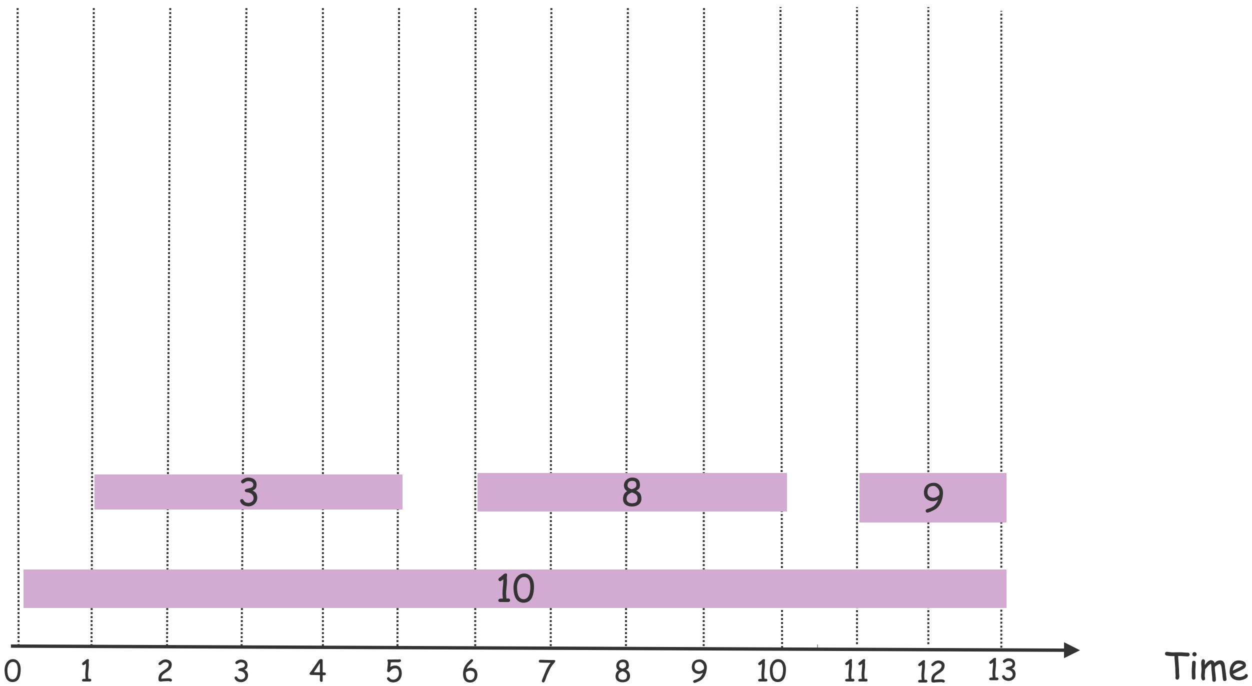
3. **结束时间早的优先：**
   依据$f_1 \le f_2 \le \dots \le f_n$ 依次向后挑选出相容的活动

提示：举反例否定掉不合理的贪心策略

1. 开始时间早的优先：

依据$s_1 \le s_2 \le \ldots \le s_n$ 依次向后挑选出相容的活动

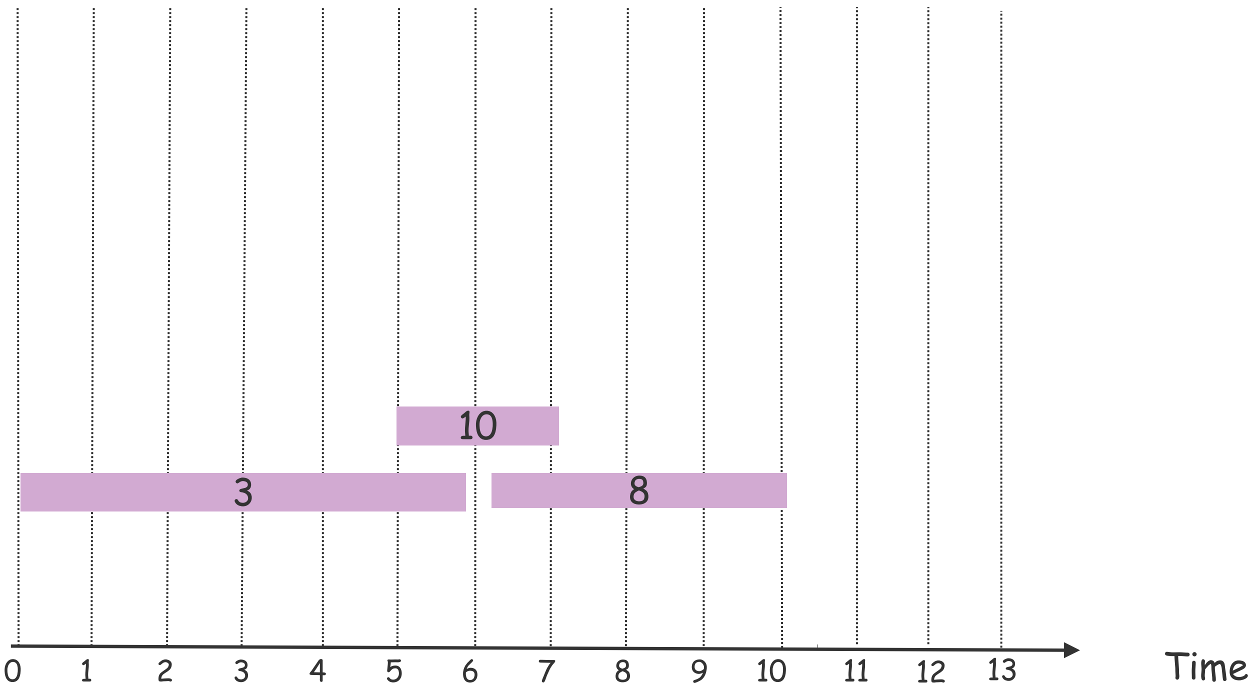
反例：



2. 占用时间短的优先：

依据$f_1-s_1 \le f_2-s_2 \le \dots \le f_n-s_n$依次向后挑选出相容的活动

反例：



贪心挑选过程是多步判断过程，每步依据某种“贪心”策略进行活动选择

活动选择策略：

1. **开始时间早的优先 (X)**

   依据 $s_1 \le s_2 \le \ldots \le s_n$ 依次向后挑选出相容的活动

2. **占用时间短的优先 (X)**

   依据 $f_1-s_1 \le f_2-s_2 \le \ldots \le f_n-s_n$ 依次向后挑选出相容的活动

3. **结束时间早的优先：**

   依据 $f_1 \le f_2 \le \ldots \le f_n$ 依次向后挑选出相容的活动

提示：举反例否定掉不合理的贪心策略

3. 结束时间早的优先：

依据$f_1 \le f_2 \le \ldots \le f_n$ 依次向后挑选出相容的活动

s和f中的活动已按$f_1 \le f_2 \le \ldots \le f_n$排序：

```
int greedySelector(int [] s, int [] f, boolean a[]) {
    int n=s.length-1;  a[1]=true;  1号活动选入
    int j=1; int count=1;       j：已选入的最后活动标号
    for (int i=2;i<=n;i++) { 从2号活动，依次考察和j号活动是否相容
      if (s[i]>=f[j]) {
        a[i]=true;
        j=i;
        count++;
        }
      else a[i]=false;
      }
    return count;
  }
```

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}, 结束 t = 11

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

如何证明对所有输入都能取得正确的解?

贪心算法特点

1. 贪心算法可用于组合优化问题

2. 贪心算法对应多步判断的过程，最终的判断序列对应于问题的最优解

3. 根据某种贪心选择性质进行判断，性质好坏决定算法的成败

4. 必须进行正确性证明取得全局最优解

5. 证明不正确的技巧：举反例

优点：算法简单、时间和空间复杂度较低

贪心算法证明

数学归纳反 + 反证法

第一数学归纳法

证明对任何自然数n，都有：

$$1+2+\cdots+n=\frac{n(n+1)}{2}$$

证明：1）n=1时，左边=1，右边=1*(1+1) / 2 = 1

2）假设对于n成立，证明对n+1成立：

1+2+...+n+n+1 = (1+2+...+n)+n+1

=n(n+1)/2 + n+1 = (n+1)(n/2+1) = (n+1)(n+2)/2

贪心算法证明

数学归纳反 + 反证法

第一数学归纳法：适合证明涉及自然数的命题P(n)

归纳基础：证明P(1）为真（或P(0)为真）

归纳步骤：假设对n有P(n)为真，证明P(n+1)为真

推理逻辑：对任意n，P(n)为真 → P(n+1)

P(1)为真

n=1, P(1） → P(2)

n=2, P(2） → P(3)

...

贪心算法证明

数学归纳反 + 反证法

第二数学归纳法：适合证明涉及自然数的命题P(n)

归纳基础：证明P(1）为真（或P(0)为真）

归纳步骤：假设对所有小于n的k有P(k)为真，

证明P(n)为真

推理逻辑：对任意k<n，P(k)为真 → P(n)

P(1)为真

n=2, P(1）→ P(2)

n=3, P(1)和P(2)为真 → P(3)

…

用数学归纳法证明贪心算法

1. 描述一个关于自然数n的命题，该命题可断定贪心策略执行将得到最优解，n可代表问题规模或算法步数

2. 证明该命题对所有自然数为真

归纳基础：从最小实例规模开始

归纳假设：第一数学归纳法 or 第二数学归纳法

命题：活动选择算法执行到第k步，已选择了k项活动 $i_1, i_2, \ldots, i_k$

则该算法存在最优解A包含上述活动 $i_1, i_2, \ldots, i_k$

证明上述命题后，到第n步即得到问题的最优解

令S={1,2,…,n}是活动集，已按$f_1 \le f_2 \le \dots \le f_n$排序

归纳基础：k=1，证明存在最优解包含活动1

设最优解A**不包含活动1**，将其中活动按截止时间排序，不妨设A中第一个活动为k（k≠1）

用活动 1 替换 A 中的活动 j 得 B

$$B = A - \{K\} \cup \{1\}$$

由于$f_1 \le f_k$，所以B也是最优解，且B中包含有活动1

$f_1 \le f_k$

k					A
1					**B**

令$S=\{1,2,\ldots,n\}$是活动集，已按$f_1 \le f_2 \le \ldots \le f_n$排序

归纳步骤：假设命题对于k为真，证明对k+1也为真

活动选择算法执行到第k步，已选择了$1,i_2,\ldots,i_k$

则根据归纳假设，一定存在最优解A包含$i_1,i_2,\ldots,i_k$活动

A中除$1,i_2,\ldots,i_k$之外，剩余活动选自集合S

A: 最优解集合

令$S=\{1,2,\ldots,n\}$是活动集，已按$f_1 \le f_2 \le \ldots \le f_n$排序

归纳步骤：假设命题对于k为真，证明对k+1也为真

A: 最优解集合

在此情况下，B是S的最优解（否则假设S的最优解为C，C的活动比B多，则$C\cup\{1,i_2,\ldots,i_k\}$是最优解，且比A的活动多，与A是最优解矛盾）

令$S=\{1,2,\ldots,n\}$是活动集，已按$f_1 \le f_2 \le \ldots \le f_n$排序

归纳步骤：假设命题对于k为真，证明对k+1也为真

A: 最优解集合

将S看成子问题，根据归纳基础，存在S上的最优解B_2包含其中第一个活动i_{k+1}，且B_2和B的活动个数一样多，于是

$$\{1,i_2,\ldots,i_k\} \cup B_2 = \{1,i_2,\ldots,i_k,i_{k+1}\} \cup B_2 - \{i_{k+1}\}$$

也是原问题的最优解

六、典型算法

6.1 贪心算法

6.1.1 贪心思想

6.1.2 活动选择案例

6.1.3 装载问题案例

6.1.4 其他案例

将n个集装箱1,2,...,n装上货轮，集装箱i的重量为w_i，货轮载重上限为C（$w_i \le C$），无体积限制，如何使得货轮装载的集装箱最多?

建模分析：设$<x_1,x_2,\ldots,x_n>$为解向量，$x_i=1$当且仅当第i个集装箱装上船

如何设计贪心算法? 轻者优先!

算法设计：将集装箱按重量排序，使得：

$$w_1 \le w_2 \le \ldots \le w_n$$

按标号从小到大装箱，直到下个箱子使得集装箱总重量超过货轮载重量限制，则停止

正确性证明

命题：对装载问题任何规模为n的输入实例，算法得到最优解，设集装箱从轻到重记为1,2,…,n

归纳基础：只包含1个集装箱的情况，显然可得最优解

归纳步骤：对n个箱子的情况贪心法都能得到最优解，

希望证明：对任何n+1个箱子的情况也能得到最优解

$$N=\{1,2,\ldots n+1\},\ w_1 \le w_2 \le \ldots \le w_{n+1}$$

改造为问题规模为n的情况：拿掉1号箱子，令$C'=C-w_1$

此时$N'=\{2,\ldots n+1\}$（问题规模为n）

由归纳假设，可得N′和C′的最优解I′

六、典型算法

6.1 贪心算法

6.1.1 贪心思想

6.1.2 活动选择案例

6.1.3 装载问题案例

6.1.4 其他案例

正确性证明

在I′基础上加入箱子1，得到I，要证明I是N的最优解

反证法：若不是，一定存在包含1的关于N的最优解I*（如果I*中没有1，用1替换I*中的第一个元素也可得到最优解），且|I*| > |I|，则I*-{1}是N′和C′的最优解，且

$$|I^*-\{1\}| > |I^*-\{1\}| = |I'|$$

与I′是关于N′和C′的最优解矛盾

饼干分发问题

设g数组表示小朋友的胃口：{1,2,7,10}

设s数组表示饼干大小：{1,3,5,9}

如何让饼干尽可能满足更多小朋友的胃口?

最大子序和

给定数组，求数组中最大连续子序列的和

{-2,1,-3,4,-1,2,1,-5,4}