第4章 贪心算法

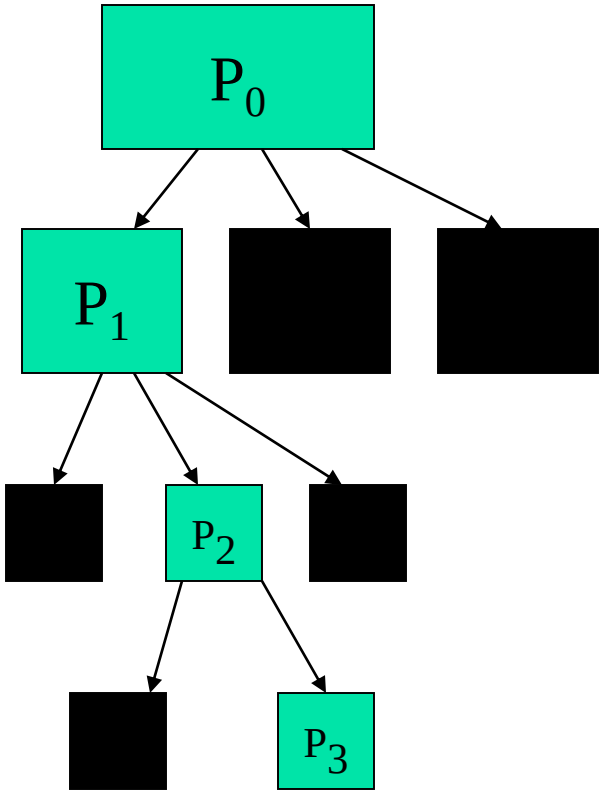
内容纲要

- 贪心算法的基本要素
- 贪心算法与动态规划的比较
- 贪心算法的应用举例
- 贪心算法的理论基础

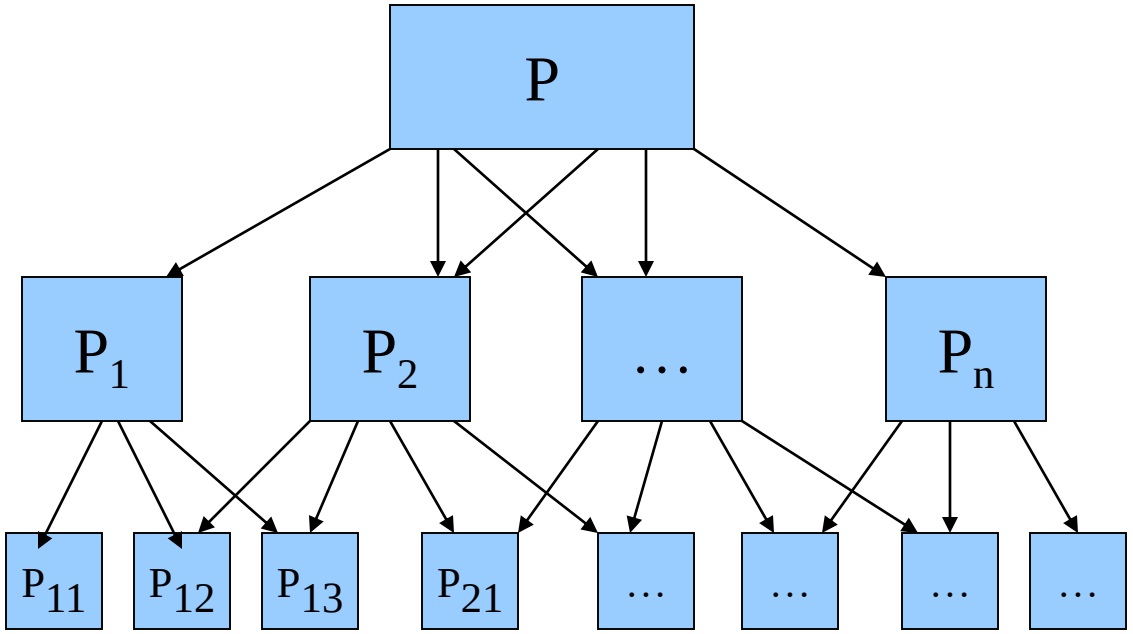
贪心算法的基本思想

- 基本特点
  - 通过一系列的选择来寻找一个问题的最优解。
  - 在当前状态下，总是选择在某种意义下的最优选择。
  - 效率高，不能保证找到最优解，但通常能找到较好的解。
- 与动态规划方法的对比
  - 共同点：问题具有最优子结构特征
  - 动态规划方法寻找问题最优解时依赖所有子问题的解
  - 贪心算法从一个局部解开始扩展，作出最优选择后得到一个新的子问题，所作的选择只依赖过去所作的选择，直到找到整体最优解。

与动态规划的对比-示意图



贪心算法产生的结构



动态规划产生的结构

4.1 活动安排问题

- 问题描述

  给定n个活动的集合E=$\{a_1, a_2, \ldots, a_n\}$,其中每个活动都要使用同一资源，但同一时间内只有一个活动能使用这一资源。每个活动a_i都要求从时间s_i开始使用，到f_i结束，且$s_i < f_i$。如果选择了a_i，则在区间$[s_i, f_i)$内占用资源。活动安排问题就是要在所给的活动集合中选择出最大的相容活动子集。

- 相容活动与相容活动集

  称活动a_i和a_j是相容的，如果$f_i \le s_j$或$f_j \le s_i$。

  如果活动集A中的任意两个活动都是相容有，则称A为相容活动集。

贪心算法求解活动安排问题

- 数据组织

用数组s和f存储活动的起始时间和结束时间，且按结束时间的非减顺序存储，即$f_1 < f_2 \ldots < f_n$。排序时间为O($n$log$n$)。

- 贪心算法思想

从a_1开始构造相容的活动A。

如果下一个活动a_{i+1}与A相容，则将a_{i+1}加入A中。

这个构造过程中，总是选择结束时间最早的相容活动，以便为以后未安排的活动留出尽可能多的时间（贪心选择）。

贪心算法求解活动安排问题

```
void GreedySelector(int n, Type s[], Type f[], bool A[]) {
      A[1]=true;
       int j=1;
      for (i=2; i<=n;i++) {
          if (s[i]>=f[j])  {
             A[i]=true;
             j=i;
          }
          else  A[i]=false;
    }
}
```

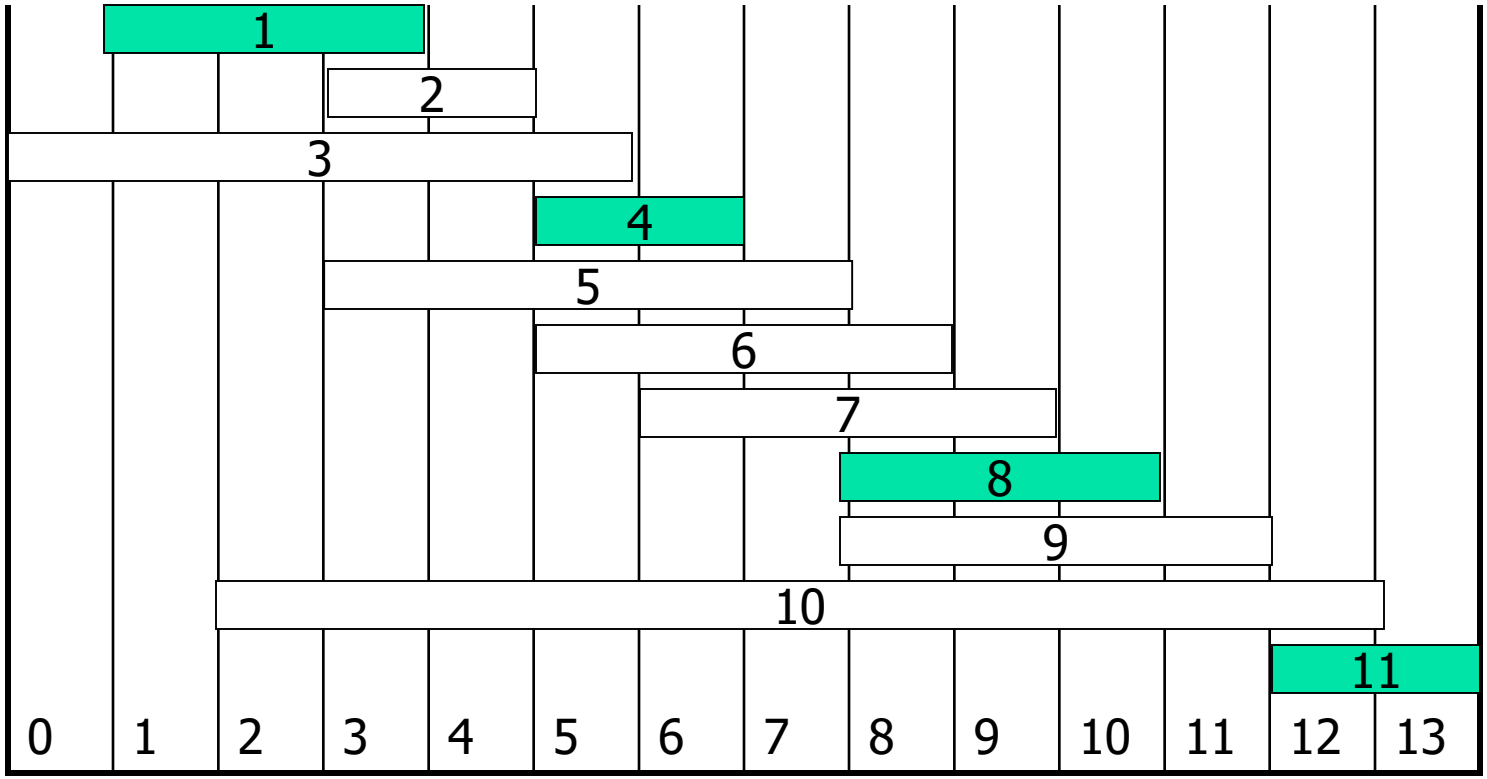
活动安排贪心算法的时间复杂性

- 时间复杂性：排序+贪心算法
- 活动排序算法的时间复杂性O(***n***log***n***)
- 贪心算法的复杂性O(***n***)

实例

- 有11个活动

i	1	2	3	4	5	6	7	8	9	10	11
s	1	3	0	5	3	5	6	8	8	2	12
f	4	5	6	7	8	9	10	11	12	13	14



活动安排贪心算法的正确性

- 总存在一个以贪心选择开始的最优活动安排方案
- 选择了活动1后，原问题化简为对E中选择所有与活动1相容的活动选择的子问题。

4.2 贪心算法的基本要素

- 贪心选择性质
  - 所求问题的最优解可以通过一系列的局部最优解的择选，即贪心选择得到。
  - 在贪心算法中，在问题当前状态下作出最优选择，然后再去解作出选择后产生一个的子问题。
  - 通常求解方向是自顶向下进行。
  - 证明贪心选择性质是关键一步。

证明贪心选择性质

- 证明技巧与结构
  - 技巧：证明每一步所作的贪心选择最终导致问题一个整体最优解
  - 结构：
    - 首先考察问题的一个整体最优解，并证明修改这个最优解，使其以贪心选择开始。说明作了贪心选择后，原问题简化为一个相同性质的规模更小的问题。
    - 然后用数学归纳法证明，通过每一步贪心选择，最终可获得问题的一个整体最优解。
- 在活动安排问题中，证明最优解可以从贪心选择a1开始。
  - 若A是对于E的活动安排问题的一个最优解，设A的第一个活动是a_k，则有两种情况：若 $k=1$，则A就是一个以贪心选择开始的最优解；若$k>1$，则设$B=A-\{a_k\}\cup\{a_1\}$。由于$f_1\leq f_k$，且A是相容活动集，故B也是相容活动集。又由于B中活动数目与A相同，故也是最优的。也就是说B是一个以贪心选择选择a_1开始在最优解。

最优子结构性质

- 当问题的最优解包含了它的子问题的最优解时，称此问题具有最优子结构性质。
- 在活动安排问题中，最优子结构性质表现为：若A是对于E的活动安排问题包含活动1的最优解，则相容活动集A’=A-{$a1$}是对于E’={$a_i \in E: s_i > f_1$}的活动安排问题的最优解。
- 最优子结构性质是贪心算法和动态规划算法的关键性质。

3.贪心算法与动态规划的差异

- 共同点：问题具有最优子结构性质。
- 算法选择：用贪心算法还是动态规划？
- 贪心算法能解的问题是否动态规划算法也能解？
- 动态规划算法能解的问题是否贪心算法也能解？
- 考察两个相似的问题

0-1背包问题

- 给定n种物品和一个背包。物品I的重量是w_i，其价值为v_i，背包的容量为c。问如何选择装入背包的物品，使得装入背包的物品的价值最大。
- 每种物品要么装入包中，要么不装入。不能部分地装入。
- 形式描述：给定c>0, w_i>0, v_i>0,1≤i≤n，要求找到一个n元0-1向量(x_1, x_2, …, x_n), x_i=0或x_i=1, 1≤i≤n，使得 $\sum_{i=1}^{n} w_i x_i \le c$ ，而且 $\sum_{i=1}^{n} v_i x_i$ 达到最大。

背包问题

- 给定n种物品和一个背包。物品I的重量是w_i，其价值为v_i，背包的容量为c。问如何选择装入背包的物品，使得装入背包的物品的价值最大。
- 每种物品可以部分地装入。
- 形式描述：给定c>0, w_i>0, v_i>0,1≤i≤n，要求找到一个n元向量$(x_1, x_2, \ldots, x_n)$, $0 \le x_i \le 1, 1 \le i \le n$，使得

$$\sum_{i=1}^{n} w_i x_i \le c$$

，而且

$$\sum_{i=1}^{n} v_i x_i$$

达到最大。

0-1背包问题与背包问题的相似性

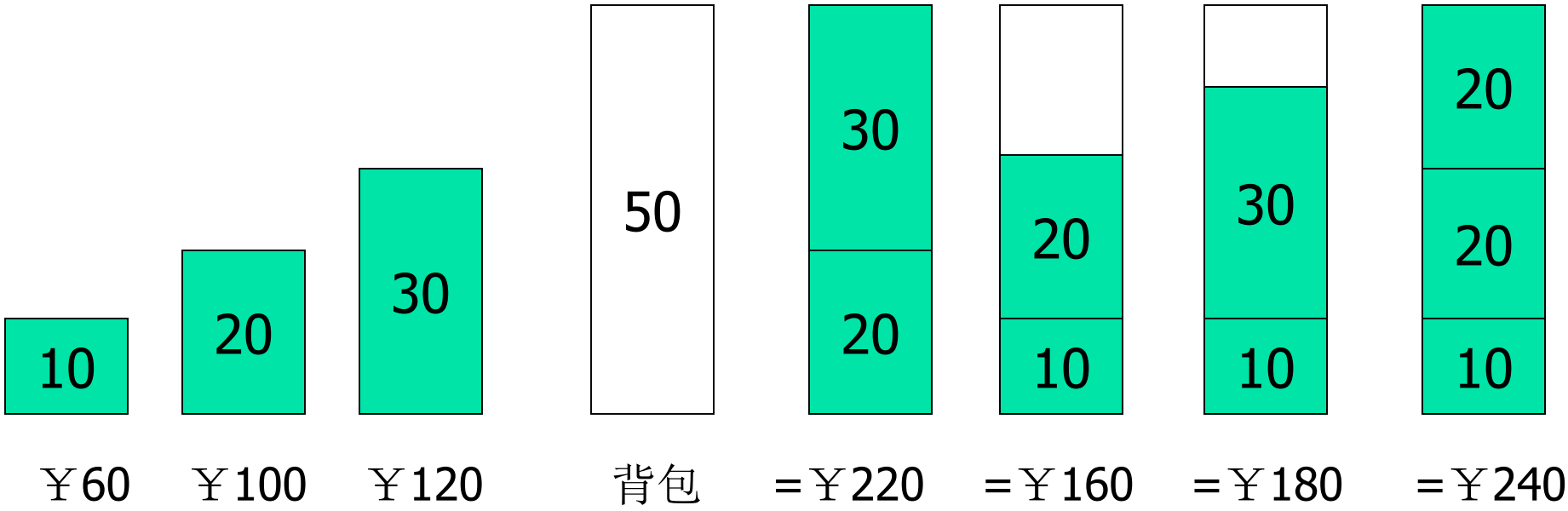
- 具有最优解子结构性质
  - 对于0-1背包问题，设A是装入背包物品的价值最大的物品集合，则$A_j=A-\{j\}$是其它n-1个物品1,2,…,j-1,j+1,…, n可装入容量为$c- w_j$ 的背包中价值最大的物品集合。
  - 相似地，对于背包问题，设A是装入背包物品的价值最大的物品集合，从A拿出物品j已装入部分（设重量为w），所剩的物品集记为A_j，则A_j是其它n-1个物品1,2,…,j-1,j+1,…, n以及重量为$w_j- w$的物品j可装入容量为$c- w$的背包中价值最大的物品集合。

贪心算法求背包问题

```
void Knapsack(int n, float M,Float v[], float w[], float x[]) {
    sort(n,v,w);
    int i=1;
    for (i=1; i<=n;i++)   x[i]=0;
    float c=M;
    for (i=1; i<=n;i++)   {
        if (w[i]>c)  break;
        x[i]=1;
        c-=w[i];
    }
    if (i<=n)   x[i]=c/w[i];
}
```

贪心算法求解0-1背包问题-反例

- 用贪心算法解3种物品的0-1背包问题，失败。
- 用贪心算法解3种物品的背包问题，成功。



4.3 最优装载问题

- 有一批集装箱要装到一艘载重量为c的轮船上。其中集装箱i的重量是w_i，。如果装载体积不受限制，问如何尽可能多地选择集装箱装船。
- 集装箱不可分拆，且船不能超载。
- 形式描述：给定$c>0$, $w_i>0$, $1\le i\le n$，要求找到一个n元0-1向量$(x_1, x_2, \ldots, x_n)$，$x_i \in \{0, 1\}$, $1\le i\le n$，使得

$$\sum_{i=1}^{n} w_i x_i \le c$$

，而且

$$\sum_{i=1}^{n} x_i$$

达到最大。

1. 贪心算法描述

```
void Loading(int x[], float w[], float c, int n) {
      sort(w);
      int i=1;
      for (i=1; i<=n;i++)
                x[i]=0;
      for (i=1; i<=n;i++)   {
            if (w[i]>c)  break;
            x[i]=1;
            c-=w[i];
      }
}
```

2. 贪心选择性质

- 设集装箱已按重量由小到大排序，即 $w_1 < w_2 < \ldots < w_n$，$(x_1, x_2, \ldots, x_n)$是一个最优解。又设

$$k = \min_{1 \le i \le n}\{i \mid x_i = 1\}$$

如果问题有解，则, $1 \le k \le n$ 。

- 如果$k=1$，则$(x_1, x_2, \ldots, x_n)$是一个满足贪心选择性质的最优解
- 如果$k>1$，取$y_1=1, y_k=0, y_i = x_i$， $1<i \le n$且$i \ne k$，则有

$$\sum_{i=1}^{n} w_i y_i = w_1 - w_k + \sum_{i=1}^{n} w_i x_i \le \sum_{i=1}^{n} w_i x_i \le c$$

因此， $(y_1, y_2, \ldots, y_n)$是也一个可行解。

另一方面，由于 $\sum_{i=1}^{n} y_i = \sum_{i=1}^{n} x_i$ 知， $(y_1, y_2, \ldots, y_n)$是一个满足贪心选择性质的最优解。

3. 最优子结构性质

- 设$(x_1, x_2, \ldots, x_n)$是一个满足贪心选择性质的最优解，则$x_1=1$，且$(x_2, \ldots, x_n)$是一个轮船载重为$c-w_1$的待装船的集装箱为$2, \ldots, n$的最优装箱问题的最优解A。