

# 贪心算法

吴世东

2009-11-12

# 本次课大纲

---

- 习题讲解
  - 1017二维装箱
  - 2054树着色

# 1017二维装箱

- 题目描述
- 输入、输出
- 题目分析
  - 解题思路
  - 实现细节
  - 性能评测

# 1017二维装箱

---

## ■ 题目描述

- 盒子：一工厂生产6种盒子，每个盒子高度相同，长宽分别是 $1*1$ 、 $2*2$ 、 $3*3$ 、 $4*4$ 、 $5*5$ 、 $6*6$ 。
- 箱子：工厂只有一种箱子来装产品，箱子高度和盒子一样，长宽是 $6*6$
- 要求：找一种方法，使用最少个数的箱子装下用户要求的产品

# 1017二维装箱

## ■ 输入、输出

- 输入：每一行输入六个数字，分别代表要装入箱子的1\*1到6\*6的盒子的个数，输入全0程序结束。

如： 0 0 4 0 0 1

7 5 1 0 0 0

0 0 0 0 0 0

- 输出：需要的箱子的个数

# 1017二维装箱

---

## ■ 题目分析

- 贪心法适用：最优解问题可以优先考虑贪心
  - 满足优化原则的组合优化问题
  - 问题求解表示成多步判断
  - 整个判断序列对应问题的最优解
  - 子序列对应子问题的最优解
- 贪心选择：直观的思考，可以从大到小依次装入
- 贪心证明：是否能够用贪心得到最优解，贪心法得到的不一定是最优解。
  - 归纳法
  - 交换论证

# 1017二维装箱

- 交换论证:

- 证明思想: 本题可以转化成找一个最优插入序列问题, 从一个最优解出发, 在不改变最优性的条件下, 转变成没有逆序插入的解.

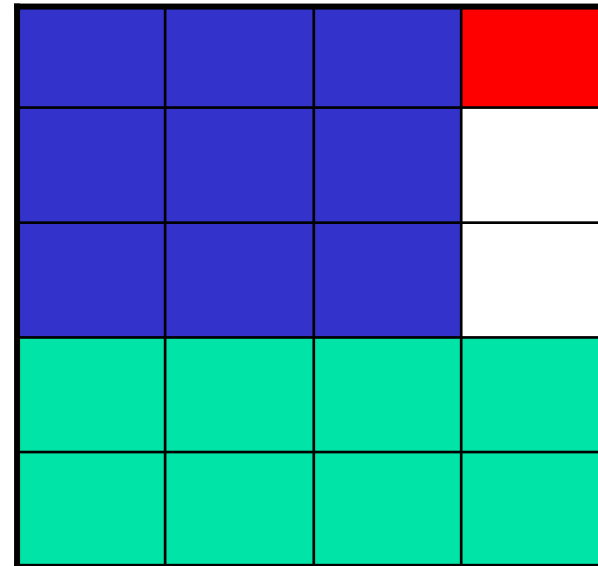
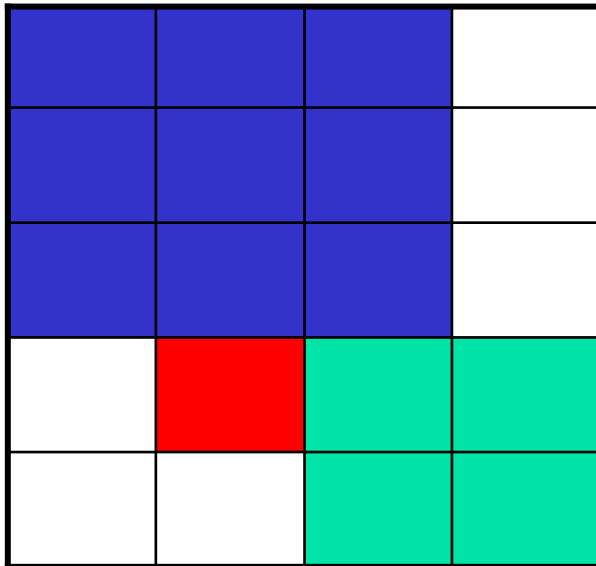
- 证明:

假设存在一个含有逆序插入的最优解 $I$ , 其中存在向一个箱子中插入盒子时, 第 $j$ 步插入的盒子的底面积 $S_j$ 小于第 $k$ 步插入的盒子的底面积 $S_k$  ( $j < k$ )。将第 $k$ 步与第 $j$ 步交换得到一个解 $I'$ , 则 $I'$ 所用的箱子数 $\leq I$ 所用的箱子数。因为如果先装入一个小的盒子 $S_j$ , 有可能会導致 $S_k$ 无法装入而新增箱子 (如下图所示), 但反过来则不会, 因此这与 $I$ 是最优解矛盾。

# 1017二维装箱

- 交换论证:

- 如果插入时存在逆序, 就可能导致箱子数增加





# 1017二维装箱

---

## ■ 解题思路:

- 使用贪心法解题，对于一个箱子，从最大的盒子开始考虑，依次在箱子的空隙中填入剩下的可以填入的盒子，直到空隙不能在满足任何剩余的盒子为止（这个过程也是从大到小在空隙中填入）。然后在进行下一个箱子填充，直到所有盒子都被填充到箱子里。
- 注意：从3\*3的盒子开始要注意从自己开始填充空隙。(trick)

# 1017二维装箱

- 实现细节:

- 数据结构: 一维数组(链表也可以)

- 伪代码: While leftboxnum!=0 do

- Chestnum++;

- Leftspace=6\*6;

- For i->boxnum do

- If box[i]<leftspace&& box[i].put==false then

- Insert(box[i]);

- box[i].put=true;

- Leftspace-box[i].space;

- Leftboxnum--;

- End if

- End For

- End While

# 1017二维装箱

---

- 性能评估:
  - 时间复杂度: 最好 $O(n)$ ,最差 $O(n)$ ,平均 $O(n)$
  - 空间复杂度:  $O(1)$

# 2054树着色

- 题目描述
- 输入、输出
- 题目分析
  - 解题思路
  - 实现细节
  - 性能评测

# 2054树着色

## ■ 题目描述:

- 树 (Bob要着色)
- 着色时间: 每个节点需要1单位时间
- 着色要求:
  - 着色完一个节点, 就可以着色下一个节点 (没有搜索时间);
  - 只有一个节点的父节点已经着色之后, 它才可以被着色 (着色从根开始)
- 着色权重:
  - $C_i$ : 节点权重
  - $F_i$ : 着色绝对时间 (从0开始计算)
- 要求: 对于一棵树, 给出其花销最小的 (  $\min \sum_{i=1}^N C_i \times F_i$  ) 着色方法

# 2054树着色

- 题目描述:

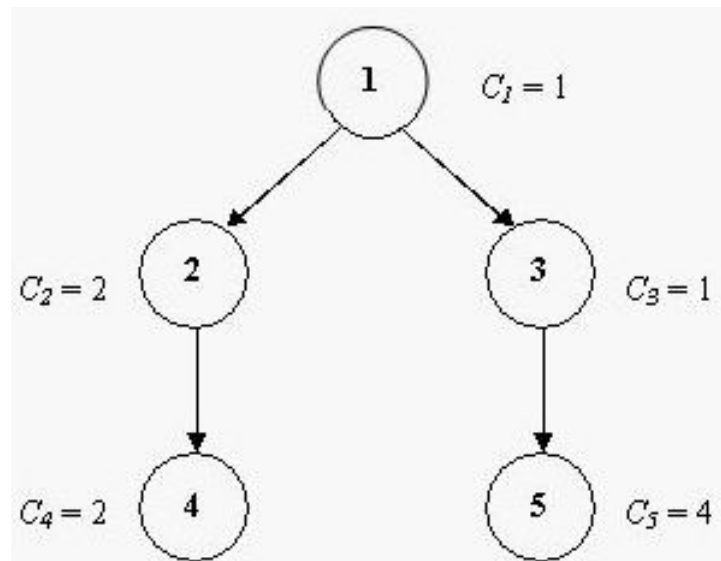


Figure-1. A tree with five nodes

# 2054树着色

## ■ 输入、输出：

### ■ 输入：

- 第一行：树的节点数目、根节点编号
- 第二行：节点的权重
- 第三行——0 0行：树枝

### ■ 输出： $\min \sum_{i=1}^N C_i \times F_i$

### ■ 例如：

### ■ Sample Input

- 5 1
- 1 2 1 2 4
- 1 2
- 1 3
- 2 4
- 3 5
- 0 0

### ■ Sample Output : 33

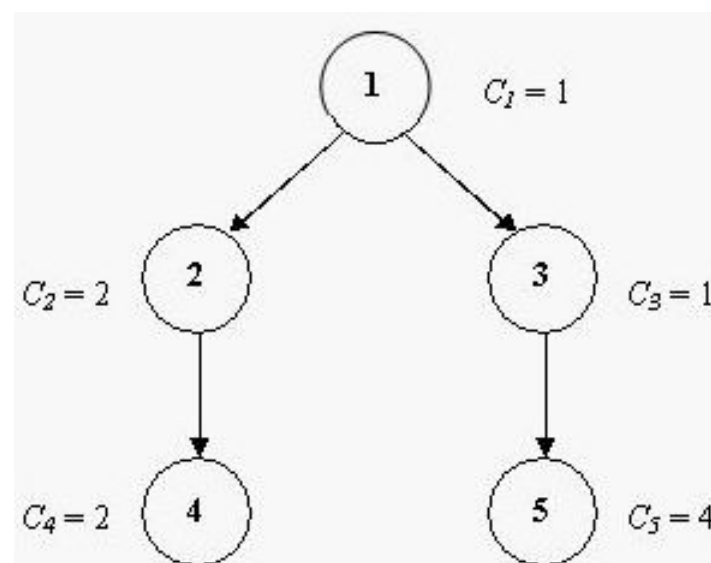


Figure-1. A tree with five nodes

# 2054树着色

## ■ 贪心证明:

- 证明思路: 由于本题实际上可以转化为求一个带权树的着色序列, 因此可以考虑使用贪心法解决, 并使用交换证明来说明使用贪心法可以得到最优解。
- 证明: 假设存在一个含有逆序的最优解 $I$  (即存在 $C_i > C_j$ , 且两节点没有父子关系, 但着色的顺序 $j > i$ , 即 $F_i > F_j$ )。那么交换两个节点的着色顺序, 得到解 $I'$ , 即 $I' = I - (C_i \times F_i + C_j \times F_j) + (C_i \times F_j + C_j \times F_i)$ 。整理一下 $I' = I - [(C_i - C_j)(F_i - F_j)]$ , 由于 $C_i - C_j > 0, F_i - F_j > 0$ , 故 $I' < I$ , 与 $I$ 是最优解矛盾。



# 2054树着色

## ■ 实现细节:

- 数据结构: 堆 (优先队列), 树 (顺序存储较好、链式存储也可)

- 伪代码:

```
Heap.insert(root);
While !Heap.empty() do
    treenode=Heap.maxC();
    treenode.color=true;
    treenode.cost=F*treenode.C
    result+= treenode.cost;
    F++;
    Pile.insert(treenode.child);
End While
```

# 2054树着色

---

- 性能评测:

- 空间复杂度:  $O(n)$
- 时间复杂度:  $O(n\log n)$  , 优先队列

# 总结

---

- 贪心法适用:

- 装箱问题
- 时间序列问题
- 任务调度问题

- 贪心法证明:

- 归纳法
- 交换论证

**Thanks !**