# 第七章 贪心法

- 概述
- 图问题中的贪心法
- 3 组合问题中的贪心法
- 4 小结



#### 找零钱问题

假设有面值为5元、2元、1元、5角、2角、1角的货币若干枚,需要找给顾客4元6角现金,为使付出的货币的数量最少。

- 1、先选择面值不超过4元6角的最大面值的货币: 2元1枚,剩余2元6角。
- 2、其次选择面值不超过2元6角的最大面值的货币: 2元1枚,剩余6角。
- 3、选择面值不超过6角的最大面值的货币: 5角1枚,剩余1角。
- 4、选择面值不超过1角的最大面值的货币: 1角1枚,剩余0角。

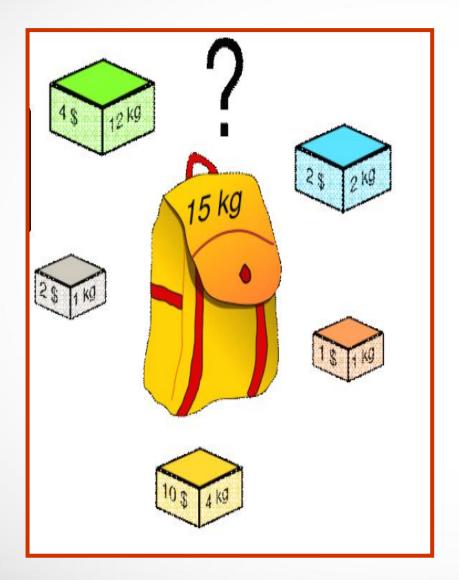
### 找零钱问题

假设有面值为3元、1元、8角、5角、1角的货币若干枚,要找给顾客4元6角现金,为使付出的货币的数量最少。

真实情况是:需要3张货币:1个3元和2个8角。

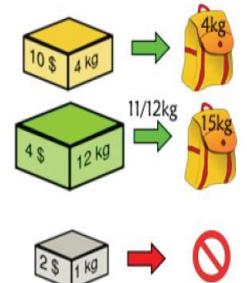
而按贪心法找给顾客的是1个3元、1个1元、 1个5角和1个1角共4张货币。

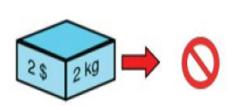
#### 背包问题

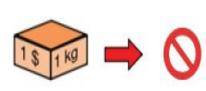


采用怎样的装包方法 才会使装入背包物品 的总收益最大? 方案1: 按物品价 值降序装包

Max=15kg



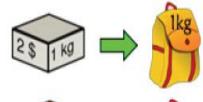


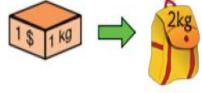


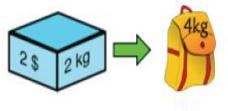
总价值: 13.67\$

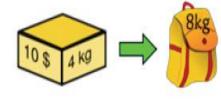
方案2: 按物品重量 升序装包

Max=15kg







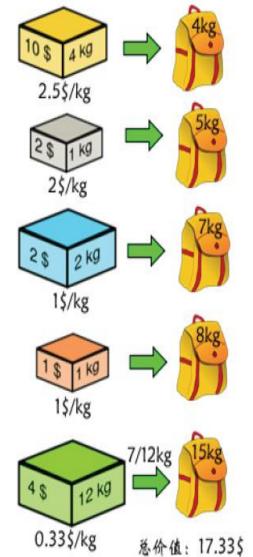




总价值: 17.33\$

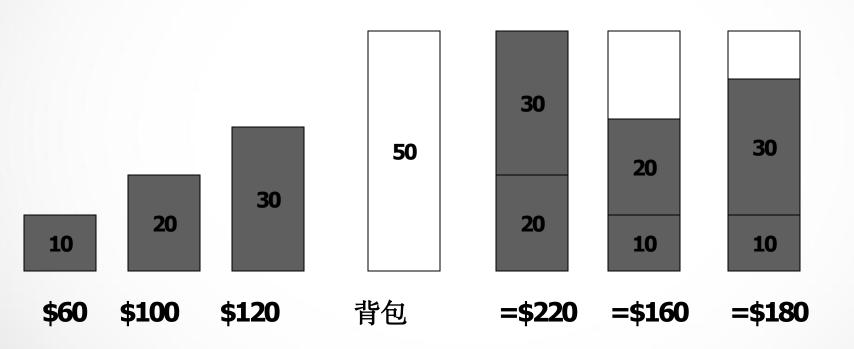
方案3: 按物品价值与重量比值的降序装包

Max=15kg



# 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。



# 贪心法的设计思想

贪心法是一种简单有效的方法。它在解决问题的策略上只根据当前已有的信息就做出选择,而且一旦做出了选择,不管将来有什么结果,这个选择都不会改变。

# 贪心法的设计思想

贪心法并不是从整体最优考虑,它所做出的选择只是在某种意义上的局部最优。

这种局部最优选择并不总能获得整体最优解,但通常能获得近似最优解。

如果一个问题的最优解只能用蛮力法穷举得到,则贪心法不失为寻找问题近似最优解的一个较好办法。

# 一个简单的例子—埃及分数

问题描述: 古埃及人只用分子为1的分数, 在表示一个真分数时, 将其分解为若干个埃及分数之和, 例如: 7/8表示为1/2 + 1/3 + 1/24。埃及分数问题要求把一个真分数表示为最少的埃及分数之和的形式。

# 埃及分数——想法

设真分数为A/B, B除以A的整数部分为C, 余数为D, 则有下式成立:

$$\mathbf{B} = \mathbf{A} \times \mathbf{C} + \mathbf{D}$$

即:

$$B/A = C + D/A < C + 1$$

则:

$$A/B > 1/(C + 1)$$

即1/(C+1)即为真分数A/B包含的最大埃及分数。

设E = C + 1, 由于

$$A/B - 1/E = ((A \times E) - B)/(B \times E)$$

则真分数减去最大埃及分数后,得到真分数

$$((A \times E) - B)/B \times E$$

该真分数可能存在公因子, 需要化简。

# 图问题中的贪心法 ——TSP问题

问题描述: TSP问题是指旅行家要旅行n个城市, 要求各个城市经历 且仅经历一次然后回到出发城市, 并要求所走的路程最短。

### 想法1: 最近邻点策略

从任意城市出发,每次在没有到过的城市中选择最近的一个,直到经过了所有的城市,最后回到出发城市。

$$C = \begin{bmatrix} \infty & 3 & 3 & 2 & 6 \\ 3 & \infty & 7 & 3 & 2 \\ 3 & 7 & \infty & 2 & 5 \\ 2 & 3 & 2 & \infty & 3 \\ 6 & 2 & 5 & 3 & \infty \end{bmatrix}$$



#### 想法2: 最短链接策略

每次在整个图的范围内选择最短边加入到解集合中,但是,要保证加入解集合中的边最终形成一个哈密顿回路。因此,当从剩余边集E'中选择一条边(u,v)加入解集合S中,应满足以下条件:

- ① 边(u, v)是边集E'中代价最小的边;
- ② 边(u, v)加入解集合S后, S中不产生回路;
- ③ 边(u, v) 加入解集合S后, S中不产生分枝。

$$C = \begin{bmatrix} \infty & 3 & 3 & 2 & 6 \\ 3 & \infty & 7 & 3 & 2 \\ 3 & 7 & \infty & 2 & 5 \\ 2 & 3 & 2 & \infty & 3 \\ 6 & 2 & 5 & 3 & \infty \end{bmatrix}$$



# 图问题中的贪心法 ——图着色问题

问题描述: 给定无向连通图G=(V,E), 求图G的最小色数k, 使得用k种颜色对G中的顶点着色, 可使任意两个相邻顶点着色不同。

# 图问题中的贪心法 ——图着色问题

考虑顶点顺序为1,2,3,4,5得到最优解 考虑顶点顺序为1,5,2,3,4得到近似解

# 图问题中的贪心法 ——图着色问题

- 1. 所有顶点置未着状态;
- 2. 颜色k初始化为0;
- 3. 循环直到所有顶点均着色
  - 3.1 取下一种颜色k++;
  - 3.2 依次考察所有顶点
    - 3.2.1 若顶点i已着色,则转步骤3.2;
    - 3.2.2 若顶点i着颜色k不冲突,则color[i]=k;
- 4. 输出各顶点的着色;

说明: color[n]表示顶点n的着色情况。

# 组合问题中的贪心法一背包问题

问题描述: 给定n种物品和一个容量为C的背包, 物品i的重量是w<sub>i</sub>, 其价值为v<sub>i</sub>, 背包问题是如何选择装入背包的物品, 使得装入背包中物品的总价值最大?



### 想法:

#### 贪心策略的选择:

- (1) 选择价值最大的物品;
- (2) 选择重量最轻的物品;
- (3) 选择单位重量价值最大的物品。









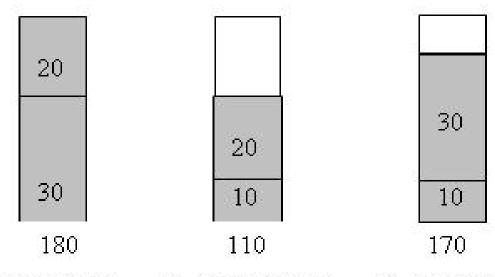




策略1: 选择价值最大的物品;

策略2: 选择重量最轻的物品;

策略3: 选择单位重量价值最大的物品。



(b) 贪心策略 1

(c) 贪心策略 2 (d) 贪心策略 3

### 背包问题算法

设背包容量为C, 共有n个物品, 物品重量存放在数组w[n]中, 价值存放在数组v[n]中, 问题的解存放数组x[n]。

- 1. 改变数组w和v的排列顺序,使其按单位重量价值v[i]/w[i]降序排列;
- 2. 将数组x[n]初始化为0;
- 3. i=1;
- 4. 循环直到(w[i]>C)
  - 4.1 将第i个物品放入背包: x[i]=1;
  - 4.2 C=C-w[i];
  - 4.3 i++;
- 5. x[i]=C/w[i];

# 组合问题中的贪心法一活动安排问题

\*设有n个活动的集合 $E=\{1,2,...,n\}$ , 其中每个活动都要求使用同一资源,而在同一时间内只有一个活动能使用这一资源。

- \*每个活动i都有一个使用该资源的起始时间 $s_i$ 和一个结束时间 $f_i$ ,且 $s_i < f_i$  。如果选择了活动i,则它在半开时间区间[ $s_i, f_i$ ]内占用资源。若区间[ $s_i, f_i$ ]与区间[ $s_j, f_j$ ]不相交,则称活动i与活动j是相容的。也就是说,当 $s \ge f_i$ 或 $s \ge f_i$ 时,活动i与活动j相容。
- \*活动安排问题要求在所给的活动集合中选出最大的相容活动子集。

### 活动安排问题——想法

贪心法求解活动安排问题的关键是如何选择贪心 策略,使得按照一定的顺序选择相容活动,并能 安排尽量多的活动。

至少有两种看似合理的贪心策略:

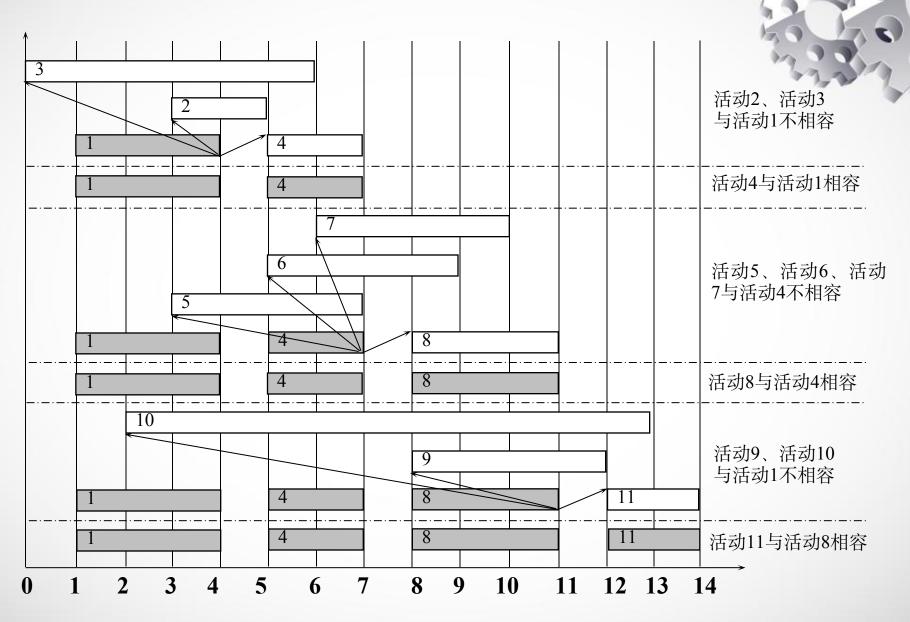
- (1)最早开始时间:可以增大资源的利用率。
- (2)最早结束时间:可以使下一个活动尽早开始。

# 活动安排问题——实例

设有11个活动等待安排,这些活动按结束时间的非减序排列如下:

i	1	2	3	4	5	6	7	8	9	10	11
$s_i$	1	3	0	5	3	5	6	8	8	2	12
$f_i$	4	5	6	7	8	9	10	11	12	13	14

# 活动安排问题——求解过程



# 活动安排问题——算法

设有n个活动等待安排,这些活动的开始时间和结束时间分别存放在数组s[n]和f[n]中,集合B存放问题的解,即选定的活动集合。

- 1. 对数组f[n]按非减序排序,同时相应地调整s[n];
- 2. 最优解中包含活动1: B={1};
- 3. j=1; i=2;
- 4. 当(i≤n)时循环执行下列操作
  - 4.1 如果(s[i]>=f[j]) 则
    - 4.1.1  $B=B+\{j\};$
    - 4.1.2 j=i;
  - 4.2 i++;

# 组合问题中的贪心法一多机调度问题

问题描述:设有n个独立的作业 $\{1,2,...,n\}$ ,由m台相同的机器 $\{M_1,M_2,...,M_m\}$ 进行加工处理,作业i所需的处理时间为 $t_i$ ( $1 \le i \le n$ ),每个作业均可在任何一台机器上加工处理,但不可间断、拆分。多机调度问题要求给出一种作业调度方案,使所给的n个作业在尽可能短的时间内由m台机器加工处理完成。

# 多机调度问题——想法

贪心法求解多机调度问题的贪心策略是最长处理时间作业优先,即把处理时间最长的作业分配给最先空闲的机器,这样可以保证处理时间长的作业优先处理,从而在整体上获得尽可能短的处理时间。

按照最长处理时间作业优先的贪心策略,当 $m \ge n$ 时,只要将机器i的 $[0,t_i)$ 时间区间分配给作业i即可;当m < n时,首先将n个作业依其所需的处理时间从大到小排序,然后依此顺序将作业分配给空闲的处理机。

# 多机调度问题——实例

设7个独立作业 $\{1, 2, 3, 4, 5, 6, 7\}$ 由3台机器 $\{M_1, M_2, M_3\}$ 加工处理,各作业所需的处理时间分别为 $\{2, 14, 4, 16, 6, 5, 3\}$ 。

$M_1$	作业4									
$M_2$										
$M_3$	作业5		作业3	,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,						
时间	<b></b> 6	> 5	<b>4</b>							
分配										

- **P146-147**:
- •T5, T6, T10
- •T5: 给定n位正整数a, 删除掉其中任意的k (k<n) 个数字后, 剩余的数字按照原来的次序排列, 组成一个新的正整数。对于给定的正整数a和k, 设计算法找出剩余数字组成的新整数最小的删数方案。
- •T6: 设有n个顾客同时等待一项服务,顾客i需要的服务时间为ti(1≤ti≤n)。设计算法,应如何安排n个顾客的服务次序,才能使得顾客总的等待时间最小。

### ■作业

- **P146-147**:
- •T5, T6, T10
- •T10: 一辆汽车加满油后可行驶n公里,旅途中有若干个加油站,加油站之间的距离由数组A[m]给出,其中A[i]表示第i-1个加油站到第i个加油站的距离,旅途的起点和终点都各有一个加油站。设计一个算法,计算沿途需要停靠加油的地方,使得加油次数最少。