# 贪心方法

长沙市第一中学
曹利国

# 贪心方法的基本思想

- 贪心是一种解题策略，也是一种解题思想
- 使用贪心方法需要注意局部最优与全局最优的关系，选择当前状态的局部最优并不一定能推导出问题的全局最优
- 利用贪心策略解题，需要解决两个问题：
  - 该题是否适合于用贪心策略求解
  - 如何选择贪心标准，以得到问题的最优解

# 适用于贪心策略求解的问题的特点

适用于贪心策略求解的问题必须具有最优子结构的性质，但并不是所有具有最优子结构的问题都可以用贪心策略求解。因为贪心往往是盲目的，需要使用更理性的方法——动态规划（例如“0-1 背包问题”与“部分背包问题”）

# 贪心方法的应用

**例题 1：** 节点网络。

现有一个 N！个节点的网络，每个节点的编号都是编号（$A_1A_2A_3\cdots A_N$）序列的一个置换。对于任意两个节点 S 和 T，如果 T 的编号是由 S 编号的首位与除首位外的编号中任一位交换所得，则 S 和 T 之间有一条边，求从给定节点 S 走到节点（$A_1A_2A_3\cdots A_N$）所需经过的最少边数。其中，$n \leqslant 100$。

# 贪心方法的应用

例如 n=3 的情况：

| | |
|---|---|
| $(A_1A_2A_3)$ | $(A_1A_3A_2)$ |
| $(A_2A_3A_1)$ | $(A_2A_1A_3)$ |
| $(A_3A_1A_2)$ | $(A_3A_2A_1)$ |

# 贪心方法的应用

**【分析】**从题意表面上看，本题是一个求最短路径的问题，但题设中的N≤100，也就是说图中最多有100！个节点，采用二维关系的图结构根本无法存贮这众多的状态。通过问题的本质分析，可以将问题转化为一个序列的最优转化问题。

# 贪心方法的应用

采用***贪心策略***：

每次让一个节点归位或为下一步工作做准备。

其具体步骤为：

- 若序列中第一个点为 $A_x$（$x \neq 1$），则将第一个点和第 x 个点交换。这便完成了让一个点归位的工作；
- 若第一个是 $A_1$，则任找一个编号与位置不相符的点，并与之交换。这样下一步便可让交换到 1 号位置的点归位。

# 贪心方法的应用

下面看一个 n=4，初始序列为（$A_3A_4A_1A_2$）的推演过程：

（$A_3A_4A_1A_2$）　第一个点为 $A_3 \neq A_1$，将第 3 个点 $A_1$ 与 $A_3$ 交换

（$A_1A_4A_3A_2$）　第一个点 $A_1$ 已归位，但第二个点为 $A_4 \neq A_2$，将第 2 个点 $A_4$ 与 $A_1$ 交换

（$A_4A_1A_3A_2$）　第一个点为 $A_4 \neq A_1$，将第 4 个点 $A_2$ 与 $A_4$ 交换

（$A_2A_1A_3A_4$）　第一个点为 $A_2 \neq A_1$，将第 2 个点 $A_1$ 与 $A_2$ 交换

（$A_1A_2A_3A_4$）　已经符合要求了

一共经过 4 步完成

。

# 贪心方法的应用

**例题 2**： d- 规则问题。

对任意给定的 $m(m\in N_+)$ 和 $n(n\in N_+)$，满足 $m<n$，构造一初始集合： $P=\{x|m\leqslant x\leqslant n,x\in N_+\}(m,n\leqslant 100)$

现定义一种 d 规则如下：若存在 $a\in P$，且存在 $K\in N_+$, $K>1$，使得 $K\times a\in P$，则修改 P 为： $P=P-\{y|y=s\times a,s\in N_+\}$，并称该 d 规则具有分值 a。现要求编制一个程序，对输入的 m,n 值，构造相应的初始集合 P，对 P 每应用一次 d 规则就累加其相应的分值，求能得到最大累加分值的 d 规则序列，输出每次使用 d 规则时的分值和集合 p 的变化过程。

# 贪心方法的应用

**【分析】**

初看这一问题，很容易想到用贪心策略来求解，即选择集合中最大的可以删除的数开始删起，直到不能再删除为止，而且通过一些简单的例子来验证，这一贪心标准似乎也是正确的，例如，当 m=2， n=10 时，集合 P＝{2,3,…,10}，运用上述“贪心标准”可以得到这一问题的正确的最优解 d=5＋4＋3＝12，即其 d- 规则过程如下：

1. a=5　P={2,3,4,6,7,8,9}　d=5
2. a=4　P={2,3,6,7,9}　d=5+4=9
3. a=3　p={2,7}　d=5+4+3=12

# 贪心方法的应用

但是，如果再仔细地分析一个例子，当 m=3 ， n ＝ 18 时，如果还是使用上述“贪心标准”，则得到问题的 d- 规则总分为 d=35 ，其 d- 规则序列为（ 9,8,7,6,5 ），而实际上可以得到最大 d- 规则总分为 d ＝ 38 ，其对应的 d- 规则序列为（ 9,8,7,6,3,5 ）。

为什么会出现这样的反例呢？这是因为，问题中要使得 d- 规则总分 d 值越大，不光是要求每一个 d 分值越大越好，也要求取得的 d 分值越多越好。

因此，本题**不能**采用纯粹的贪心策略求解。

# 贪心方法的应用

**【改进】**

将原算法基础上进行改进。下面给出新的算法：

3. 建立集合 P={m..n}
4. 从 n div 2 到 m 每数构造一个集合 c[i]，包含该数在 P 中的所有倍数（不包括 i 本身）
5. 从 n div 2 起找到第一个元素个数最少但又不为空的集合 c[i]
6. 在 d 分值中加上 i
7. 把 i 及 c[i] 集合从 P 集中删除，更新所有构造集合的元素
8. 检查所有构造集合，若还有非空集合，则继续 3 步骤，否则打印、结束

# 贪心方法的应用

下面看 m=3，n=18 时的推演过程：

2. 初始 P={2..18}

- 找到 i=9，c[i]={18}，P={3..8,10..17}
- 找到 i=8，c[i]={16}，P={3..7,10..15,17}
- 找到 i=7，c[i]={14}，P={3..6,10..13,15,17}
- 找到 i=6，c[i]={12}，P={3..5,10,11,13,15,17}
- 找到 i=3，c[i]={15}，P={4,5,10,11,13,17}
- 找到 i=5，c[i]={10}，P={4,11,13,17}

到此所有构造集合全部为空， d=9+8+7+6+3+5=38

# 贪心方法的应用

**讨论：**

- 能否证明此贪心策略是正确的？
- 能否找到其他更好的算法？

# 贪心方法的应用

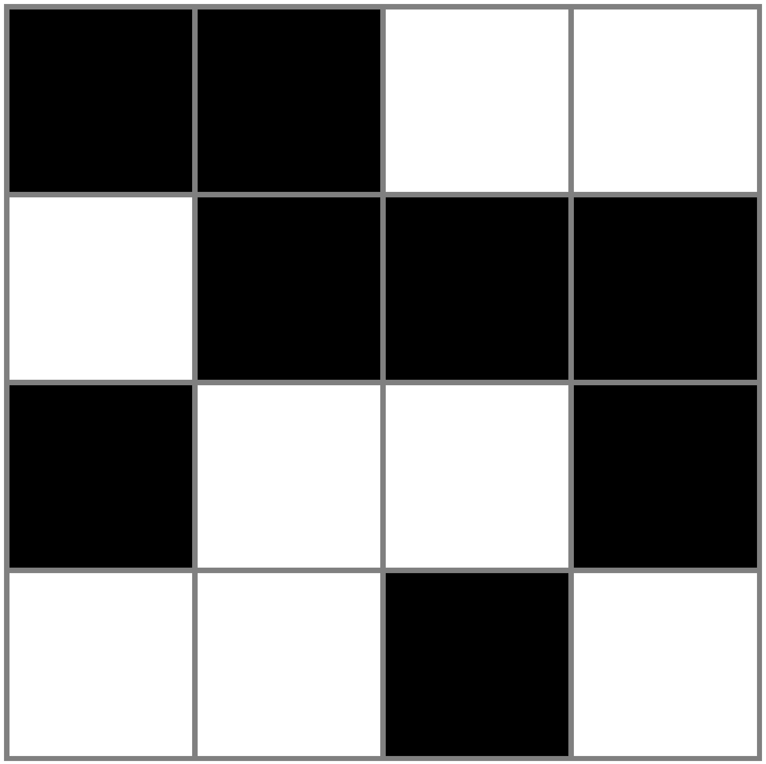
**例题 3**： 射击竞赛

射击的目标是一个由 $R\times C(2\leqslant R\leqslant C\leqslant 1000)$ 个小方格组成的矩形网格。每一列恰有 2 个白色的小方格和 R-2 个黑色的小方格。行从顶至底编号为 1~R，列从左至右编号为 1~C 。射击者可射击 C 次。

在连续的 C 次射击中，若每列恰好有一个白色的方格被射中，且不存在无白色方格被射中的行，这样的射击才是正确的。

如果存在正确的射击方法，则要求找到它。

# 贪心方法的应用



射击的选择有 $2^C$ 种，符合要求的却很少。要解决问题，还需从正确的射击方法的特征入手。

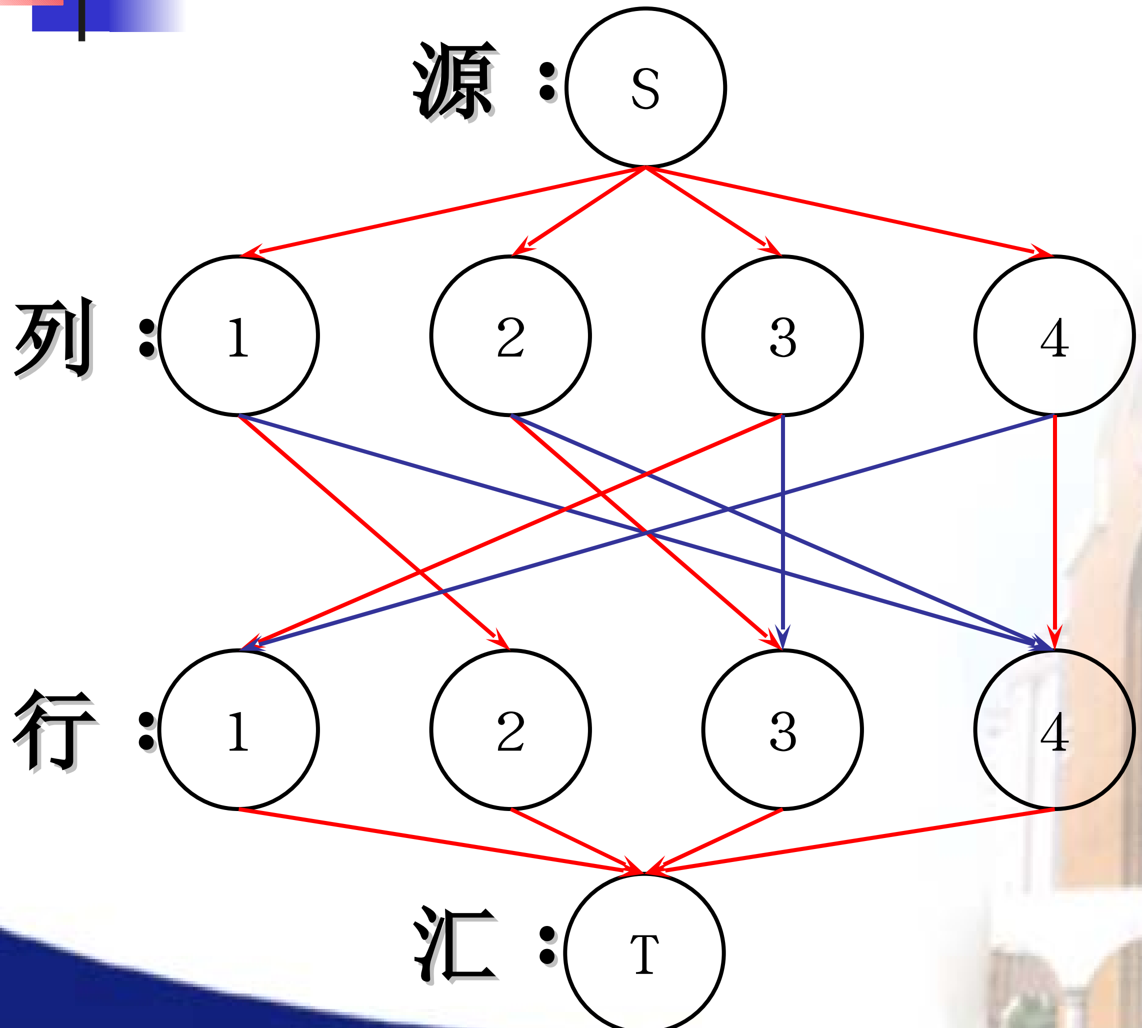
# 贪心方法的应用

**【方法一】网络流算法**

我们将表示列的点编号为 1 到 C，表示行的点编号为 C+1 到 C+R，如果一个白色方格处在第 i 行第 j 列，那么从点 j 向点 C+i 连一条弧，弧的容量为 1 。再增设一个源点 S，从点 S 往点 1 到 C 间各连一条弧，弧的容量为 1，又设一个汇点 T，从点 C+1 到点 C+R 向汇点 T 连一条弧，弧的容量为 1，那么从源点 S 到汇点 T 求最大流，求出的最大流量即为最多可以射击到的行数。各条流的路线则描述了具体的射击方案。

可以看出，如果用网络流求出的最大流量比 R 小，则问题无解，否则我们可以先根据网络流的结果求出该二分图的具体匹配方案。

# 贪心方法的应用




**红色**的连线流量为1

**蓝色**的连线流量为0

选择的射击格即为：

（1,3），（2,1），（3,2），（4,4）

# 贪心方法的应用

网络流算法经过优化，时间复杂度可以达到 $O(C\times(n+4C+4R))$

上述网络流算法虽然可以通过全部数据，但编程复杂度很高，而且极易出错，一不小心就会因为一个小错误影响整个程序。

# 贪心方法的应用

**【方法二】贪心方法**

2. 统计所有行包含的白格数
3. 从还没有射击格的行中选出一个白格数最少的
4. 检查所选的行
   - 若所选行的白格数为 0，则输出无解；
   - 否则从所选行的白格中任选一个作为射击格，并将与该格同列的另一个白格所处行的白格数减 1

- 返回到第 2 步，直到所有的行都有射击格。
- 若还有列没有选射击格，则在该列任选一白格作为射击格即可

# 贪心方法的应用

上面的贪心方法非常高效：

时间复杂度为 $O(R\times C)$，如果在程序中使用堆优化，时间复杂度将降为 $O(R\times \log_2 C)$。空间复杂度是线性的，而且非常容易实现。

现在关键的问题是——如何证明它的正确性？

# 贪心方法的应用

**【证明】**

用 h[i] 表示第 i 行的白格数。如果最开始的时候：

- min{h[i]}=0：第 i 行已经没有办法找到可作为射击格的白格，那么问题只能无解。
- min{h[i]}=1：那么第 i 行的这一个白格必须要作为射击格，否则将因第 i 行没有射击格而造成问题无解。
- min{h[i]}≥2：那么在这一行任选一个白格，顶多只会造成剩余行中有一行 h 值为 1，再处理那一行，最多也只会再造成剩余行中有一行 h 值为 1，如此往复，将保持 h 值为 1 的行数不超过 1 行，最后最坏的情况也是造成最后一行的 h 值为 1，继续下去所有行就都已选取了射击格了。

因此，如果原问题有解，该贪心方法一定能找到一种正确的方案。

由此可以证明，此贪心方法是正确的。

# 贪心方法的应用

**例题 4：** Transversal

有一个 $(2n+1)\times(2n+1)$ 的矩阵，每个单元格中有符号“+”或“-”。

定义一种取反操作：将 1 至 2n+1 这 2n+1 个整数任意排列，得到序列 $\{A_1,A_2,\cdots,A_{2n+1}\}$，然后将 $(1,A_1),(2,A_2),\cdots,(2n+1,A_{2n+1})$ 这 2n+1 个单元格中的符号取反。

求一种操作组合，使得在完成求得的操作组合后，表中“+”的个数不超过 2n 个。（$n\le 20$）

| | | |
|---|---|---|
| + | + | + |
| + | + | - |
| - | + | - |

# 贪心方法的应用

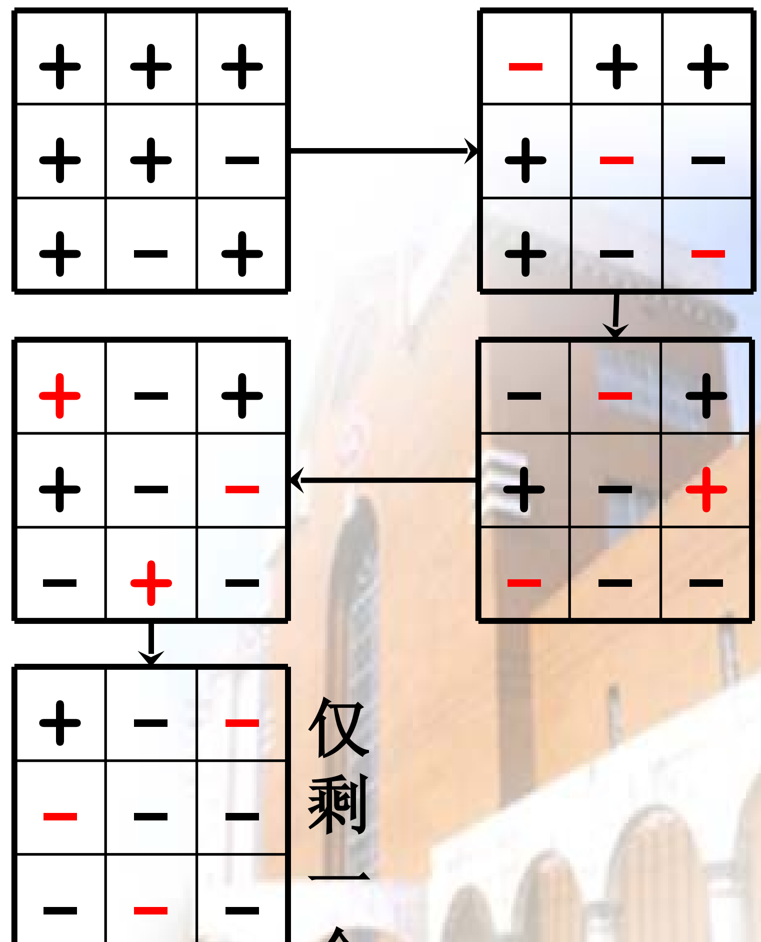
一种操作组合：

((1,1), (2,2), (3,3)),

((1,2), (2,3), (3,1)),

((1,1), (2,3), (3,2)),

((1,3), (2,1), (3,2)),



仅剩一个“+

# 贪心方法的应用

**讨论：**

是否可以用贪心法解决此题？

# 贪心方法的推广

- 贪心与其它算法结合
  - 搜索的最优化剪枝（NOI'99 生日蛋糕）
  - 优化动态规划（HNOI'99 Peter 的快餐店）
- 贪心方法与解题策略
  - 最优方法不一定是最好方法（新俄罗斯方块）
  - 想不到最优解法就用较优解法

# 贪心与其它算法结合

**例题 5：** Peter 的快餐店（***贪心与动态规划***）

Peter 最近在 R 市新开了一家快餐店。 该快餐店准备推出一种套餐，每套由 A 个汉堡、 B 个薯条和 C 个饮料组成。为了提高产量， Peter 引进了 N 条生产线。所有生产线都可以生产汉堡、薯条和饮料，由于每条生产线一天能工作的时间是有限的、不同的，而汉堡、薯条和饮料的单位生产时间又不同， Peter 需要知道，怎样安排才能是一天中生产的套餐量最大。假设一天中汉堡、薯条和饮料的产量均不超过 100 个，且生产线总数小于等于 10 。

# 贪心与其它算法结合

**【分析】**

用 p1 、 p2 、 p3 分别表示汉堡、薯条和饮料的单位生产时间， t[i] 表示第 i 条生产线每天的生产时间， p[i,j,k] 表示用前 i 条生产线生产 j 个汉堡、 k 个薯条的情况下，最多能生产的饮料数， r[i,j,k] 表示用第 i 条生产线生产 j 个汉堡、 k 个薯条的情况下，最多能生产的饮料数，则

$$p[i,j,k]=\max\{p[i-1,j1,k1]+r[i,j-j1,k-k1]\}$$

$$((j-j1)\times p1+(k-k1)\times p2<t[i])$$

通过对该算法的时间复杂度分析，最坏的情况下时间复杂度将达到 $10^9$ ，是相当费时的

# 贪心与其它算法结合

现在加入***贪心方法***，用贪心方法作预处理 ：

- 首先计算出一天生产套数的上限值：min{100 div A,100 div B,100 div C}
- 接着，用贪心方法计算出这 N 条生产线可以生产的套数，并与上限比较，大于或等于则输出上限值并退出，否则再调用动态规划。因为贪心方法的代价很低，这里甚至可以使用多次贪心标准不同的贪心方法，取其最大值。
- 在运行动态规划的过程中，也可以每完成一阶段工作便与上限值进行比较，将贪心思想充分融入到动态规划过程中，这样以来，便可望在动态规划完成前提前结束程序。

# 贪心方法与解题策略

**例题 6：** 新俄罗斯方块

新俄罗斯方块的基本规则与传统的有不同：

- 容纳基块的容器高度没有限制
- 玩家可以自己选择使用哪一种基块（与传统的俄罗斯方块一样，共有 19 种，每种都由 4 个小方块构成，包括了所有变化）
- 选取摆放的位置不能使得摆放后任意小方格悬空或超过两边的界限
- 最后的任务是使得所有列上的小方格数都为 0

# 贪心方法与解题策略

**【解法】**

这道题使用的**贪心方法**非常简单：

给每种基块打一个分（下表面越宽的分值越高），每次找到能恰好填补最低处又符合规则的分值最高的方块填上即可。

题目的要求恰好允许用贪心方法实现，因为题目没有要求用最少的步骤数，只规定了一个步数范围。

# 贪心方法与解题策略

虽然这道题已知的最优方法是分治法，而且这种贪心策略还没有办法证明其正确性。但在竞赛中，贪心方法实现简单，又能达到一定的正确率，因而不失为一种好方法。

尤其是在以拿分为目的的竞赛中，作为一种解题策略，贪心方法是很有意义的。

# 贪心方法与解题策略

近年来，近似算法的题目越来越多，比如：……

贪心方法作为一种较优算法，在这种类型的题目中是不容忽视的。

# 贪心方法小结

贪心作为一种解题思路，虽然有时无法证明它的正确性，但在无法找到其他算法的时候，不失为一种好方法。并且，贪心与其他算法的结合，可以对其他算法起到优化作用。