

贪婪的动态规划

—— 浅谈贪心思想在动态规划中的应用

绍兴县柯桥中学 黄劲松



引言

□ 在动态规划的解题中我们面临着两大困难

➤ 1、不知道是否可以用动态规划求解

➤ 2、直观的动态规划算法过于低效

□ 在这个时候，巧妙的使用贪心思想，将其融入到动态规划中，动态规划便焕发出了新的光彩

状态过于庞大
转移困难

目录

吉祥如意

◆ 贪心思想在动态规划中的应用

■ 确立状态

➤ [例一] 青蛙的烦恼（详见论文）

吉祥如意 ➤ [例二] The Horse Racing

吉祥如意 ■ 优化算法

吉祥如意 ➤ [例三] 石子归并（详见论文）

吉祥如意 ➤ [例四] The Lost House

吉祥如意

吉祥如意



贪心思想在动态规划中的应用一：确立状态

- ✓ 动态规划当中，状态的确立是重点
- ✓ 而在实际的解题过程中，状态信息往往是**隐含**的
- ✓ 这个时候，合理的运用贪心思想，可以迅速的从繁芜丛杂的问题背景中巧妙地抽象出状态

[例二]The Horse Racing

- 齐王和田忌各派出 N 匹马 ($N \leq 2000$)
- 每匹马都有一个固定的速度值
- 每场比赛，输的一方将要给赢的一方 200 两黄金，如果是平局的话，双方都不必拿出钱
- 请你扮演一下孙膑，帮助田忌赢最多的钱



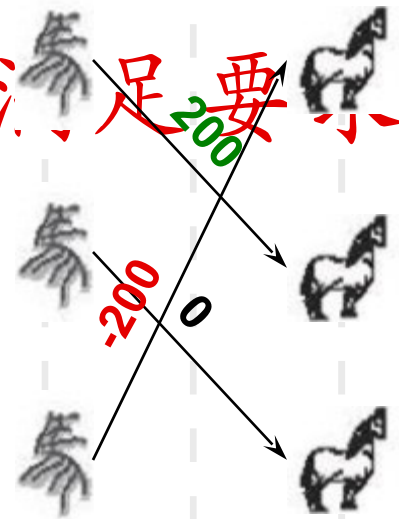
[例二] The Horse Racing

- 这个问题很显然可以转化成二分图最佳匹配的问题
- 把田忌的马放左边，把齐王的马放右边

时间复杂度过大，无法满足要求

- 如果田忌的马胜，则连一条权为 200 的边；
- 如果平局，则连一条权为 0 的边；
- 如果输，则连一条权为 -200 的边。

田忌 齐王



[例二] The Horse Racing

➤ 运用贪心思想分析问题：

- 田忌掌握有比赛的“**主动权**”，他总是根据齐王所出的马来分配自己的马去对抗齐王的马
- 可以假设齐王按照马的强弱顺序由强到弱出马

田忌最强的马 **轲 轶** 齐王最强的马

用田忌最强的马去打平齐王最强的马
用田忌最强的马去**轲 轶**齐王最强的马或者



[例二] The Horse Racing

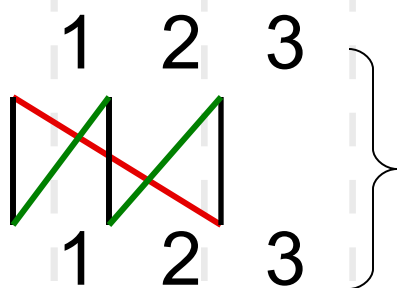
- 最强的马战平时，单一的贪心策略存在反例
- 光是打平比赛



➤ 田忌的马



➤ 齐王的马



收益为 0
200



[例二] The Horse Racing

- 最强的马战平时，单一的贪心策略存在反例
- 光是输掉比赛



➤ 田忌的马

2 3



➤ 齐王的马

1 3

收益为 0
200



[例二] The Horse Racing

- “田忌出马不是出最强的，就是出最弱的”
- 用 $f[i,j]$ 表示齐王出了 i 匹较强的马和田忌的 j 匹较强的马， $i-j$ 匹较弱的马比赛之后，田忌所能够得到的最大盈利。

$$f[i,j] = \max \{ f[i-1,j] + g[n - (i-j) + 1, i], f[i-1,j-1] + g[j,i] \}$$

$O(n^2)$

■ 其中 $g[i,j]$ 表示齐王和田忌的马分别按由强到弱的顺序排序之后，田忌的第 i 匹马和齐王的第 j 匹马赛跑所能取得的盈利，胜为 200，负为 -200，平为 0。



小结 1



✓ 抛弃了原本直观而低效的算法

✓ 结合贪心思想分析问题

✓ 用合理的假设得到了“田忌出马不是出最强，就是出最弱”的信息

✓ 因此得知可以用动态规划求解且确立出动规状态



贪心思想在动态规划中的应用二：优化算法

- ✓ 一些题目虽然容易确立出状态以及轻松的写出状态转移方程，但是直观上的算法往往效率不高
- ✓ 而贪心历来是与 **高效** 一词密不可分的
- ✓ 运用好贪心思想能够使原来效率低下的算法得到重生

[例四] The Lost House

- 蜗牛从根结点出发开始寻找它遗失在某个叶子结点的房子

- 一些中间结点上住着的虫子会告诉蜗牛它的房子是否在这条路上

树上的结点个数为 n ，最多为 1000，每个结点的分叉数 k 最多为 8。

房子遗失在每个叶子结点的概率都是相等的

■ 求蜗牛找到房子的最小数学期望步数（走过一条边算作一步）

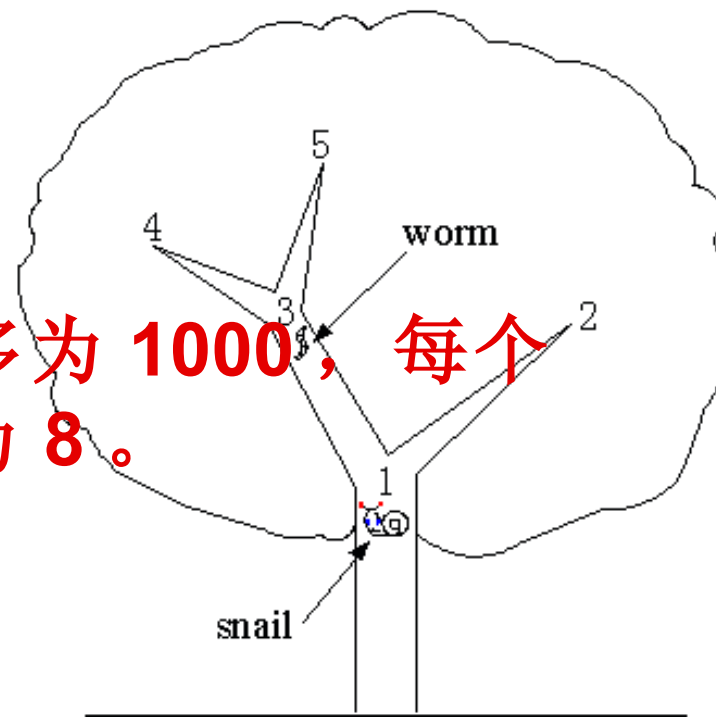
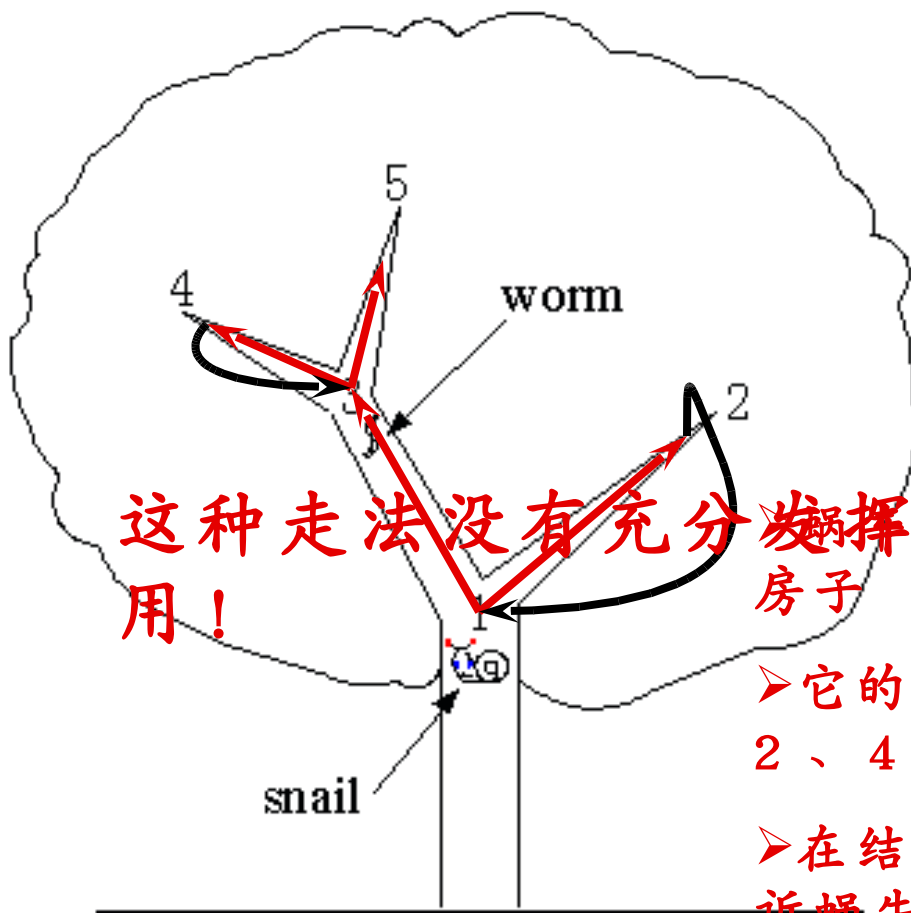


Figure-1

[例四] The Lost House



$$(1+4+6)/3 = 11/3$$

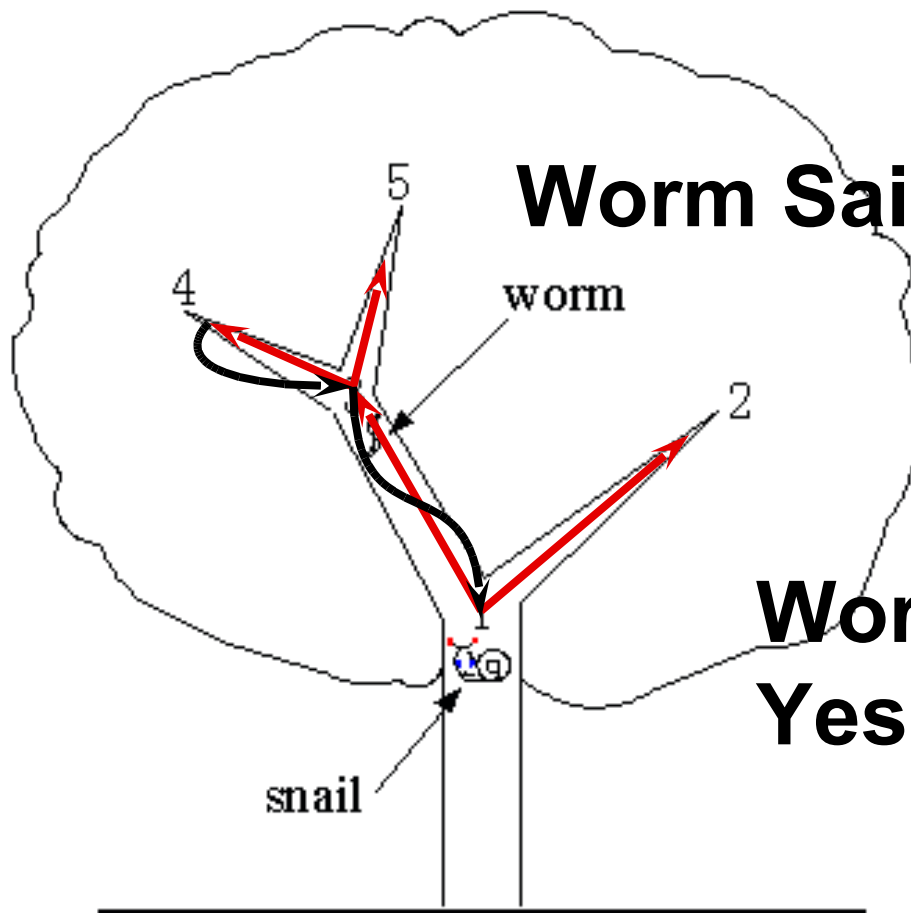
这种走法没有充分发挥由根结在这里的始作找它的房子

➤ 它的房子可能遗失在叶结点 2、4、5 上

➤ 在结点 3 上住着一只虫子，它会告诉蜗牛，房子是否在以 3 为根的子树上

Figure-1

[例四] The Lost House



Worm Said : No!

$$(3+2+4)/3 = 3$$

**Worm Said :
Yes!**

Figure-1



[例四] The Lost House

如果结论不出，本题是用树型动态规划来求解。 $S[k]$ 表示访问表示蜗牛的房子为根的子树上的期望步数和。 $Fa[u]$ 表示蜗牛的房子不在 i 为根的子树上的期望步数。 $Fb[i]$ 表示蜗牛的房子不在 i 为根的子树上的期望步数。 $for i$ 和 1 到 k 也就是遍历该子树需要的时间，如果 i 处有虫子 $begin$ 那么 $Fb[i]=0$)

$Fa[u] = Fa[S[i]] + Fb[u]$ 为根的子树上的期望步数和。 $Leaves[S[i]]$ 表示以 $S[i]$ 为根的子树上的叶子节点数。

$Fb[u] = Fb[S[i]] + 2$ 问题的解答就是 $Fa[根结点] / Leaves[根结点]$ 。

end;

$$Fa[u] = \sum_{i=1}^k \left(\sum_{j=1}^{i-1} ((Fb[S_j] + 2) + 1) \times Leaves[S_i] + Fa[S_i] \right)$$

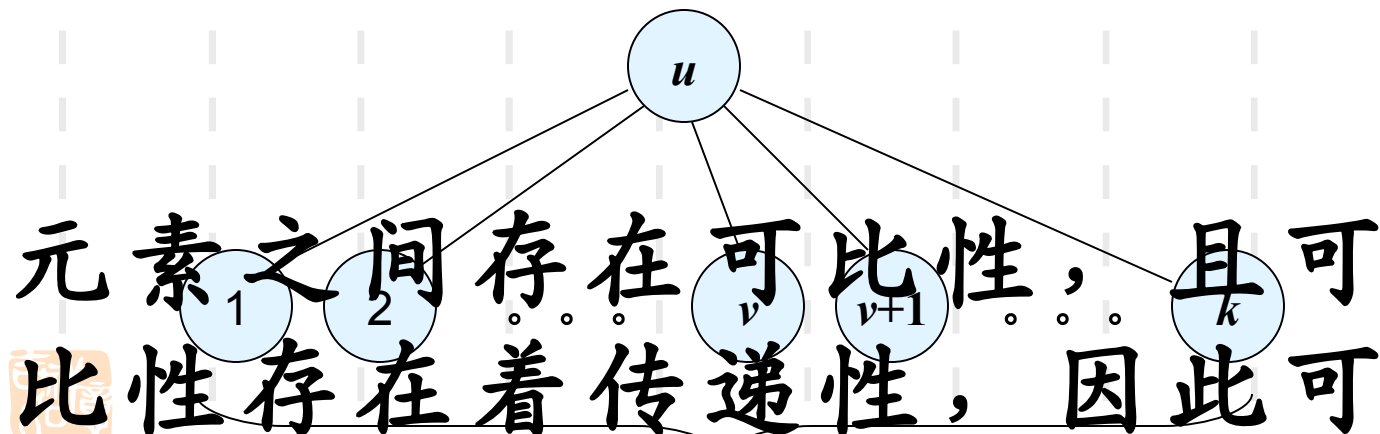
$Fb[u]$ 的值与访问顺序无关

[例四] The Lost House

- 问题的关键是如何决定儿子的访问顺序
- 一种直观的方法是枚举所有可能访问顺序，复杂度是 $O(nk!)$ ，实在是很低效
- 上述算法存在冗余，我们再用一次动态规划的话，可以将复杂度降为 $O(n2^k k)$ ，勉强可以接受了

[例四] The Lost House

► 运用贪心思想分析问题



以确定一个全序关系

这跟元素 v 和 $v+1$ 本身的信息有关！
 因此如果 A 一定在 B 前， B 一定放在 v 的前面
 所以推导出 A 一定放在 C 前。

$O(n \log n)$

小结 2



✓ 从原始的动态规划入手

✓ 运用贪心思想除去算法中的冗余

✓ 最终达到优化算法的目的



回顾与总结



➤ 贪心思想在动态规划中的两种简单应用

确立状态



优化算法



合理的运用题目中隐含
的特殊信息

回顾与总结

吉祥如意

勇于探索

大胆创新

举一反三



Thank you for listening!



欢迎提问