动态规划的优化

唐文斌

动态规划的优化

- 动态规划的时间复杂度:
  - 状态数目 * 状态转移时间复杂度

- 优化方向:
  - 减少状态数目
  - 提高状态转移效率

减少状态数目

- WC2007 疯狂赛车(racing)

提高状态转移效率

- 最长上升子序列
- Fibonacci Subsequence(ZJU2672)
- Painting the balls (SGU 183)
- USACO DEC04 divide
- 石子合并问题
- Morse Code(SPOJ 108)
- NOI2005 瑰丽华尔兹(adv1900)

最长上升子序列

- 问题
  - 给一个长度为N的整数序列A。找到A的一个最长的子序列：$x_{1..k}$ 满足
  - $1 \le x_1 < x_2 < x_3 .. < x_k \le N$
  - 且 $A[x_1] < A[x_2] < \ldots < A[x_k]$
- 状态 $F[i]$ 表示以$A[i]$结束的最长上升子序列的长度
- 状态转移
  - $F[i] = \max\{F[j]: j<i, A[j]<A[i]\} + 1$

优化

- 原动态规划算法的时间复杂度 $O(N^2)$
  - 状态 $O(N)$
  - 单个状态转移 $O(N)$
- [方向]降低状态转移时间复杂度
- 再看一眼状态转移方程
  - $F[i] = \max\{F[j]: j<i, A[j]<A[i]\} + 1$
- 瓶颈在于max，条件
  - $j < i$:　在已经计算的F中
  - $A[j]<A[i]$：另一个需要考虑的条件

优化(cont'd)

- 在一个关键字的集合中, 高效的完成
- 查询
  - 每一个关键字(f, x)
  - X在范围(-∞ , A[i])中， f最大
- 添加
  - 一个新的关键字(f, x)
- 线段树是理想的选择！
  - 转移时间复杂度$O(N) \rightarrow O(\log_2 N)$
  - 总时间复杂度$O(N^2) \rightarrow O(N\log_2 N)$

另一种优化方法

- 注意在已经有的关键字中
  - (f_1, x_1)和(f_2, x_2)
  - 若$f_1 \geq f_2$, $x_1 \leq x_2$，则(f_2, x_2)不必存在
  - 删去所有这样不必存在的关键字，维护一个新的数据结构
  - *f*严格递增, *x*严格递增
- [查询]寻找x比x_i小中最大的关键字：二分
- [维护]需要加入一个新的关键字(f, x)

ZJU2672

- 一个序列$a_1,a_2...a_n$称为Fibonacci序列如果满足$a_i = a_{i-1} + a_{i-2}$
- 给定一组整数序列, 求其最大的Fibonacci子序列($N \le 3000$)
- 10
- 1 1 3 -1 2 0 5 -1 -1 8

动态规划

- 状态:
  - F[i , j]表示以A_i与A_j结尾的最长Fib子序列长度
- 状态转移:
  - $F[i , j] = \max\{ F[k , j] + 1 \mid A_k + A_i = A_j \}$
- 时间复杂度O(N^3)
  - 状态数目：O(N^2)
  - 转移复杂度：O(N)

优化

- [方向]优化状态转移
  - $F[i, j] = \max\{ F[k, j] + 1 \mid A_k + A_i = A_j \}$
  - 只有最大的k是有意义的
- 寻找最大的k
  - 二分 : $O(\log N)$
  - 线性方法,维护顺序表 : $O(1)$

Painting the balls (SGU 183)

- 一行中有N个球, 最初都是白颜色
- 可以选择一些球染成黑色
  - 选择染黑第i个球的代价为C_i
- 要求：任意连续M个球中至少有2个黑色球
- 求代价总和最小的染色方案

- 规模
  - $2 \le N \le 10\ 000$
  - $2 \le M \le 100 \quad M \le N$

例子

- $N = 6, \quad M = 3$
- 1 5 6 2 1 3
- 最小代价为9

动态规划

- 状态信息：最后两个黑球
- $F[i, j] = c_j + \min \{F[k, i]$
- $: j - M < k < i\}$
- 时间复杂度
  - 状态数：$O(NM)$
  - 单个转移复杂度： $O(M)$
  - 总时间复杂度$O(NM^2)$

优化决策

- $F[i, j] = c_j + \min\{F[k, i]$
- $: j - M < k < i\}$

- 注意min和其求值区间
- 这里决策其实就是求一段区间中的最小值
- 使用线段树，决策时间降为$O(\log_2 M)$

- 有没有更好的办法

优化决策

- 观察：对于相同的*i*和不同的*j*
  - *******i j
  - ******i j
  - *****i   j
  - ****i   j
  - ***i   j
  - **i   j
  - *i   j
  - ****表示F[*i*, *j*]需要检查的位置

优化决策

- 结论
  - 动态规划时以*i*为阶段
  - *j*从大到小依次算出所有的F[*i*, *j*]
  - 一般来说, F[*i*, *j*]可以用到F[*i*, *j*+1]的信息
  - 这样决策的时间复杂度降为O(1)

  - 总时间复杂度O(*NM*)

USACO DEC04 Divide

- 在一段长为L(L≤10^6)的花丛中安装喷头
  - 每一个喷头的半径在A~B之间(1≤A≤B≤10^3)
  - 每个位置被且仅被一个喷头覆盖
- 有N(N≤10^3)个给定的区域, 在一个区域内的花要被同一个喷头覆盖
- 问最少需要安装多少喷头

```
            |-----c2----|-c1|
|---1---|-------2-------|---3---|
+---+---+---+---+---+---+---+---+
0   1   2   3   4   5   6   7   8
```

动态规划

- 状态:
  - F[*i*]表示最后一个喷头在*i*位置结束，至少需要多少喷头
- 状态转移:
  - $F[i] = 1 + \min\{ F[j] :$
    $2A \le i - j \le 2B$ 且*j*点不属于某个给定的区域}
- [优化方向]: 加速状态转移

优化状态转移

- 若j点属于某个给定区间
  - 不妨令$F[j] = +\infty$
  - $F[i] = 1 + \min\{ F[j] : 2A \le i - j \le 2B \}$
- 区间最小值查询
  - 线段树
  - 转移复杂度： $O(\log L)$
- 有没有更好的方法？

优化状态转移

- 若存在x_0 , x_1均满足 “$2A \le i - j \le 2B$”
  - 若 $x_0 < x_1$ 且 $F[x_0] \le F[x_1]$
  - $F[x_0]$没有价值

- 维护有序队列
- 转移复杂度: $O(1)$
- 总时间复杂度: $O(L)$

关于四边形不等式

- 动态规划状态为F[i,j]
- 决策k与F[i,k]和F[k,j]有关:设之为$k[i,j]$
- 尝试假设有如下决策单调性

$$k[i,j-1] \le k[i,j] \le k[i+1,j]$$

- 如果如此,则枚举决策k的总次数为O(N^2)

[例]石子合并问题

- 优化前：$O(N^3)$
- $F[i, j] = sum[i, j] +$
- $\min\{F[i, k] + F[k + 1, j]$
- $: i \le k < j\}$
- 优化后：$O(N^2)$
- $F[i, j] = sum[i, j] +$
- $\min\{F[i, k] + F[k + 1, j]$
- $: k[i,j-1] \le k < k[i+1,j]\}$