

Accept

你若不想做，你会找个借口；你若想做，你会找个办法。

博客园 首页 新闻 新随笔 联系 管理 订阅

昵称：Accept  
园龄：5年1个月  
粉丝：45  
关注：3  
+加关注

<2012年7月>

日	一	二	三	四	五	六
24	25	26	27	28	29	30
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	1	2	3	4

搜索

找找看

谷歌搜索

常用链接

我的随笔  
我的评论  
我的参与  
最新评论  
我的标签  
更多链接

随笔档案

2012年10月 (6)  
2012年9月 (5)  
2012年8月 (7)  
2012年7月 (1)  
2012年4月 (1)  
2012年1月 (1)  
2011年11月 (21)  
2011年9月 (1)  
2011年8月 (9)  
2011年7月 (7)

最新评论

1. Re:斜率优化DP  
上凸转180度不变成下凸了，博主没写错，注意图中 $k < j < i$ 。  
--BlueMandora  
2. Re:斜率优化DP  
既然排除了 $g[i,j] < g[j,k]$ ，那就是 $g[i,j] > g[j,k]$ ，也就是 $i$ 的斜率要大于 $j$ 的斜率，那应该是下凸。  
--Awner  
3. Re:当我真正理解了扩展欧几里得定理  
@词藻n-m不可能是负的，题目中有说哦...  
--光速小子  
4. Re:斜率优化DP  
应该是下凸  
--chenjunjie1994  
5. Re:斜率优化DP  
至于等号。。我感觉是这样吧， $dp[i]$ 的时候， $g[q[head+1],q[head]] \leq s[i]$ 时， $q[head+1]$ 不会比 $q[head]$ 差，于是 $q[head]$ 自然不要，因为斜率不严格递减， $q[h$   
 $e$ .....  
--呵呵凹凸曼

阅读排行榜

用单调队列优化DP，写给自己

单调队列是一种严格单调的队列，可以单调递增，也可以单调递减。队首位置保存的是最优解，第二个位置保存的是次优解，ect。。。

单调队列可以有两个操作：

1、插入一个新的元素，该元素从队尾开始向队首进行搜索，找到合适的位置插入之，如果该位置原本有元素，则替换它。

2、在过程中从队首删除不符合当前要求的元素。

单调队列实现起来可简单，可复杂。简单的一个数组，一个head，一个tail指针就搞定。复杂的用双向链表实现。

用处：

1、保存最优解，次优解，ect。

2、利用单调队列对dp方程进行优化，可将 $O(n)$ 复杂度降至 $O(1)$ 。也就是说，将原本会超时的N维dp降优化至N-1维，以求通过。这也是我想记录的重点

是不是任何DP都可以利用单调队列进行优化呢？答案是否定的。

记住！只有形如  $dp[i]=\max/\min (f[k]) + g[i]$  （ $k < i$  &&  $g[i]$ 是与 $k$ 无关的变量）才能用到单调队列进行优化。

优化的对象就是 $f[k]$ 。

通过例题来加深感受

<http://www.acm.uestc.edu.cn/problem.php?pid=1685>

我要长高

Description

韩父有N个儿子，分别是韩一，韩二...韩N。由于韩家演技功底深厚，加上他们间的密切配合，演出获得了巨大成功，票房甚至高达2000万。舟子是名很有威望的公知，可是他表面上两袖清风实则内心阴暗，看到韩家红红火火，嫉妒心遂起，便发微薄调侃韩二们站成一列时身高参差不齐。由于舟子的影响力，随口一句便会造成韩家的巨大损失，具体亏损是这样计算的，韩一，韩二...韩N站成一排，损失即为 $C * (韩i与韩i+1的高度差 (1 \leq i \leq N))$ 之和，搞不好连女儿都赔了。韩父苦苦思索，决定给韩子们内增高（注意韩子们变矮是不科学的只能增高或什么也不做），增高1cm是很容易的，可是增高10cm花费就很大了，对任意韩i，增高Hcm的花费是 $H^2$ 。请你帮助韩父让韩家损失最小。

Input

有若干组数据，一直处理到文件结束。 每组数据第一行为两个整数：韩子数量 $N(1 \leq N \leq 50000)$ 和舟子系数 $C(1 \leq C \leq 100)$  接下来N行分别是韩i的高度( $1 \leq hi \leq 100$ )。

首先建立方程，很容易想到的是， $dp[i][j]$ 表示第 i 个儿子身高为 j 的最低花费。分析题目很容易知道，当前儿子的身高花费只由前一个儿子影响。因此，

$dp[i][j]=\min(dp[i-1][k] + abs(j-k)*C + (x[i]-j)*(x[i]-j))$ ；其中 $x[i]$ 是第i个儿子原本的身高

我们分析一下复杂度。

首先有N个儿子，这需要一个循环。再者，每个儿子有0到100的身高，这也需要一维。再再者，0到100的每一个身高都可以有前一位儿子的身高0到100递推而来。

file:///C:/Users/wkl31415926/OneDrive/IO\_wkl\_/E5%85%A5%E9%97%A8%E9%A2%98%E7%9B%AE%E6%95%B4%E7%90%86/%E5%8A%... 1/4

- 1. 最长公共上升子序列 LCIS(13020)
- 2. 斜率优化DP(9801)
- 3. 当我真正理解了扩展欧几里得定理(6972)
- 4. 用单调队列优化DP，写给自己(4731)
- 5. 最小路径覆盖问题值得注意的地方(3204)

评论排行榜

- 1. 斜率优化DP(17)
- 2. 当我真正理解了扩展欧几里得定理(14)
- 3. 线段树专辑—— hdu 1542 Atlantis(9)
- 4. 最长公共上升子序列 LCIS(2)
- 5. 求两个超级大数的最大公约数(2)

推荐排行榜

- 1. 当我真正理解了扩展欧几里得定理(8)
- 2. 用单调队列优化DP，写给自己(4)
- 3. 最小路径覆盖问题值得注意的地方(4)
- 4. 线段树专辑—— hdu 2871 Memory Control(1)
- 5. 半平面交，求解多边形内核(1)

所以朴素算法的时间复杂度是O(n^3)。题目只给两秒，难以接受！

分析方程：

当第 i 个儿子的身高比第 i-1 个儿子的身高要高时，

$$dp[i][j]=min(dp[i-1][k] + j * C - k * C + X); \quad (k \leq j)$$
其中  $X=(x[i]-j)*(x[i]-j)$ 。

当第 i 个儿子的身高比第 i-1 个儿子的身高要矮时，

$$dp[i][j]=min(dp[i-1][k] - j * C + k * C + X); \quad (k \geq j)$$

对第一个方程，我们令  $f[i-1][k]=dp[i-1][k]-k*C$ ， $g[i][j]=j*C+X$ ；于是  $dp[i][j] = \min (f[i-1][k]) + g[i][j]$ 。转化成这样的形式，我们就可以用单调队列进行优化了。

第二个方程同理。

接下来便是如何实现，实现起来有点技巧。具体见下

```
1 #include<iostream>
2 #include<string>
3 #include<stdio.h>
4 #include<memory.h>
5 using namespace std;
6 #define inf 0xffffffff
7 #define min(a,b) a<b?a:b
8 #define max(a,b) a>b?a:b
9
10 int dp[2][101];
11 int n,c;
12 int q[101];
13 int head,tail,cur;
14
15 int main()
16 {
17     int i,j,x,nowf;
18     freopen("D:\\in.txt","r",stdin);
19     while (scanf("%d%d",&n,&c)==2)
20     {
21         scanf("%d",&x);
22         cur=0;
23         for(i=0;i<x;i++)
24             dp[cur][i]=inf;
25         for(i=x;i<=100;i++)
26             dp[cur][i]=(x-i)*(x-i);
27         for(i=1;i<n;i++)
28         {
29             scanf("%d",&x);
30             cur=1-cur;
31             //比前一个人高
32             head=tail=0;
33             for(j=0;j<=100;j++) //当身高为j时候，队列里便已经保存了0~j-1的信息，注意，是第i-1
34                 //个人的信息
35                 {
36                     nowf=dp[1-cur][j]-j*c;
37                     while(head<tail && q[tail-1]>nowf)
38                         tail--;
39                     q[tail++]=nowf;
40                     if(j<x)
41                         dp[cur][j]=inf;
42                     else
43                         dp[cur][j]=q[head]+j*c+(x-j)*(x-j);
44                 }
45             //比前一个人矮
46             head=tail=0;
47             for(j=100;j>=0;j--) //当身高为j时候，队列里便已经保存了100~j+1的信息，正写反写是有
48                 //技巧的
49                 {
50                     nowf=dp[1-cur][j]+j*c;
51                     while(head<tail && q[tail-1]>nowf)
```

```

52         if(j>=x)
53             dp[cur][j]=min(dp[cur][j],q[head]-j*c+(x-j)*(x-j));
54     }
55 }
56 int ans=inf;
57 for(i=0;i<=100;i++)
58     ans=min(ans,dp[cur][i]);
59 printf("%d\n",ans);
60 }
61 return 0;
62 }

```

还有一个比较适合理解该优化方法的题目是HDU 3401<http://acm.hdu.edu.cn/showproblem.php?pid=3401>

大概题目便是：一个人知道接下来T天的股市行情，想知道最终他能赚到多少钱。

构造状态 $dp[i][j]$ 表示第i天拥有j只股票的时候，赚了多少钱

状态转移有：

1、从前一天不买不卖：

$$dp[i][j]=\max(dp[i-1][j],dp[i][j])$$

2、从前i-W-1天买进一些股：

$$dp[i][j]=\max(dp[i-W-1][k]-(j-k)*AP[i],dp[i][j])$$

3、从i-W-1天卖掉一些股：

$$dp[i][j]=\max(dp[i-W-1][k]+(k-j)*BP[i],dp[i][j])$$

这里需要解释一下为什么只考虑第i-W-1天的买入卖出情况即可。想想看，i-W-2天是不是可以通过不买不卖将自己的最优状态转移到第i-W-1天？以此类推，之前的都不需要考虑了，只考虑第i-W-1天的情况即可。

对买入股票的情况进行分析，转化成适合单调队列优化的方程形式

$$dp[i][j]=\max(dp[i-W-1][k]+k*AP[i]-j*AP[i]). \text{ 令 } f[i-W-1][k]=dp[i-W-1][k]+k*AP[i], \text{ 则 } dp[i][j]=\max(f[i-W-1][k]-j*AP[i]).$$

这便可以用单调队列进行优化了。卖股的情况类似分析。

[View Code](#)

最后再说一个应用，用单调队列来优化多重背包问题 hdu 2191

如果有n个物品，每个物品的价格是w，重量是c，且每个物品的数量是k，那么用这样的一些物品去填满一个容量为m的背包，使得得到的背包价值最大化，这样的问题就是多重背包问题。

对于多重背包的问题，有一种优化的方法是使用二进制优化，这种优化的方法时间复杂度是 $O(m*\sum \log k[i])$ ，具体可以见

<http://www.cnblogs.com/ka200812/archive/2011/08/06/2129505.html>

而利用单调队列的优化，复杂度是 $O(mn)$

首先，对于第i件物品，如果已知体积为V，价值为W，数量为K，那么可以按照V的余数，将当前的体积V分成V组(0,1,...,V-1)。

对于任意一组，可以得到转移方程： $f[i*V+c]=f[k*V+c]+(i-k)*W$ ，其中c是V组分组中的任意一个

令 $f[i*V+c]=dp[i]$ ，那么就得到 $dp[i]=dp[k]+(i-k)*W (k \geq i-K)$

+

View Code

好文要顶

关注我

收藏该文

Accept

关注 - 3

粉丝 - 45

+加关注

40

(请您对文章做出评价)

« 上一篇：[实现类对串行化的支持](#)  
» 下一篇：[斜率优化DP](#)

posted @ 2012-07-11 11:25 Accept 阅读(4731) 评论(1) 编辑 收藏

抱歉！发生了错误！麻烦反馈至contact@cnblogs.com

[刷新评论](#) [刷新页面](#) [返回顶部](#)

注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#)，[访问网站首页](#)。

- 最新IT新闻：**
- 科学家发现银河系周围存在数百个隐藏星系
  - Groupon第四季度净亏损4650万美元 同比转亏
  - 三星Galaxy S7或配备大容量电池提升续航能力
  - 梅耶尔推翻其振兴计划 雅虎裁员大幕已开启
  - 美科学家发现系外行星与地球内部结构相似
- » [更多新闻...](#)

- 最新知识库文章：**
- 学习新东西的唯一方法
  - 关于烂代码的那些事（下）
  - 每个架构师都应该研究下康威定律
  - 消息队列（Message Queue）基本概念
  - 编程每一天（Write Code Every Day）
- » [更多知识库文章...](#)