DP优化杂谈

——斜率优化基础

By nodgd

特别行动队 NKOJ2215

【题目简述】

· 给你一个长度为n的正整数序列$\{x_1, x_2, \dots, x_n\}$，你要将其分成若干段。如果一段的和为x，则这一段的收益为

$$F(x) = ax^2 + bx + c$$

其中a, b, c为三个输入的参数。你要使总收益最大，求最大总收益。

【数据范围】

· $20\%: n \le 1000$

· $50\%: n \le 10000$

· $100\%: n \le 10^6, -5 \le a \le -1, |b| \le 10^7, |c| \le 10^7, 1 \le x_i \le 100$

· 每个测试点时间限制1秒，空间限制128MB

谈谈你的看法

考虑DP：

- 用$f[i]$表示前i个数分成若干段的最优值，写出状态转移：
- $f[i]=\max\limits_{1\le j\le i-1} f[j]+F\left(\sum_{k=j+1}^{i} x_k\right)$
- 需要化简，用前缀和处理x_k，设$s_i=s_{i-1}+x_i$
- 回忆以前学的单调队列优化DP，需要将右边的部分分离成i,j相互独立的两个部分，然后对与j相关的部分使用单调队列。
- 尝试将右边分离为i,j独立的两部分：
- $f[i]=\max\limits_{1\le j\le i-1} f[j]+F\left(s_i-s_j\right)$

$$=\max_{1\le j\le i-1} f[j]+a\left(s_i-s_j\right)^2+b\left(s_i-s_j\right)+c$$

$$=a{s_i}^2+bs_i+c+\max_{1\le j\le i-1} f[j]+a{s_j}^2-bs_j-2as_is_j$$

- 状态转移方程中出现了s_is_j这样的项，用一般的式子变形无法将其分离，这个时候怎么办呢？

这就要用到斜率优化了！

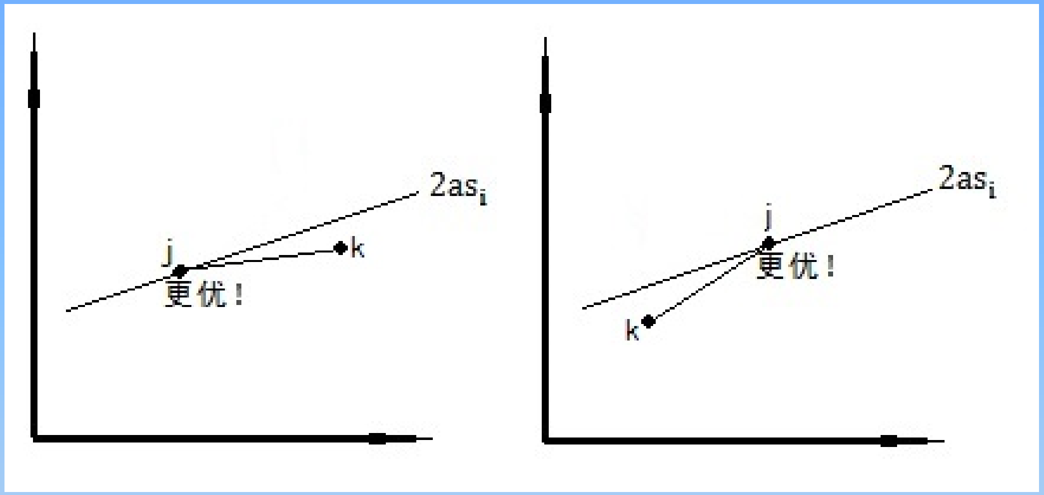
- 式子化不动了？没关系，自己创造式子！
- 假设$j<k$，且由$j\to i$比$k\to i$更优，我们得到与之等价的表达式：
- $f[j]+as_j{}^2-bs_j-2as_is_j>f[k]+as_k{}^2-bs_k-2as_is_k$

- 我们把上式中与i相关的部分放到式子一边，其他放到另一边：
- $2as_i\left(s_k-s_j\right)>(f[k]+as_k{}^2-bs_k)-\left(f[j]+as_j{}^2-bs_j\right)$

- 再令$u_j=f[j]+as_j{}^2-bs_j$，得到：
- $2as_i>\dfrac{u_k-u_j}{s_k-s_j}$
- 注意这时式子的右边，如果我们把(s_j,u_j)看做平面上一个点，不妨称为j号点，同理有k号点。则我们得到结论：
- 当$j<k$时，$\mathrm{j}\to i$比$k\to i$更优等价于j号点和k号点之间的斜率小于$2as_i$。
- 同理我们可以得到当$j>k$时对应的等价条件

我们怎样利用这个关于斜率的结论呢？

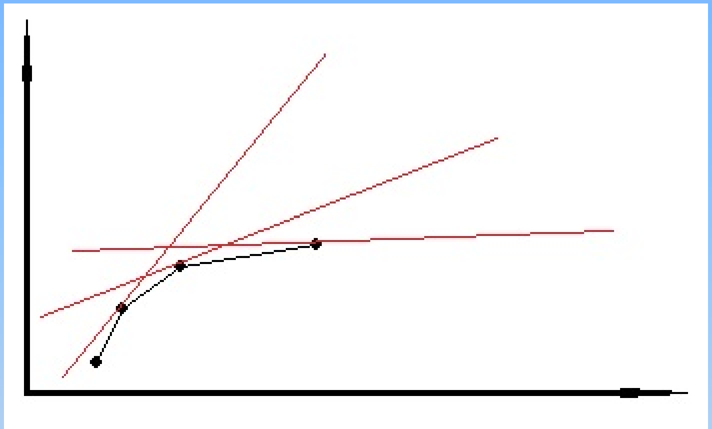
- 把j号点和k号点的图像画出来：



- 这让我们想到，可用状态一定是所有状态的上凸包。
- 当我们需要进行状态转移时，只需要去凸包上找一条斜率为$2as_i$的切线即可。
- 因为凸包上加点操作只在最右边进行，所以这个凸包可以用一个栈来维护，转移状态时在栈中二分查找切线的切点。
- 这样做的复杂度是$O(n\log n)$，仍然不能通过本题。

进一步优化

- 观察我们需要去凸包查找切线的斜率$2as_i$，我们发现，它是单调递减的。
- 又注意到，用栈维护的上凸包上的每个点处的切线的斜率也是单调递减的，如图



- 所以，最优转移的位置也是单调递增的。也就是说，如果$j \to i$是关于i的最优转移，$j' \to i + 1$是关于$i + 1$的最优转移，那么一定有$j \le j'$
- 既然这样，我们可以进行进一步优化。不用栈而改用双端队列来维护凸包，新点加入时先从队尾删除所有不满足凸性质的点，然后从队尾加入；状态转移时从队首暴力删掉不是切点的状态，直到队首是切点时停止并转移状态。
- 这样一来，每个状态都恰好进队一次出队一次，总的时间复杂度是$O(n)$的，可以通过本题。

放出伪代码：

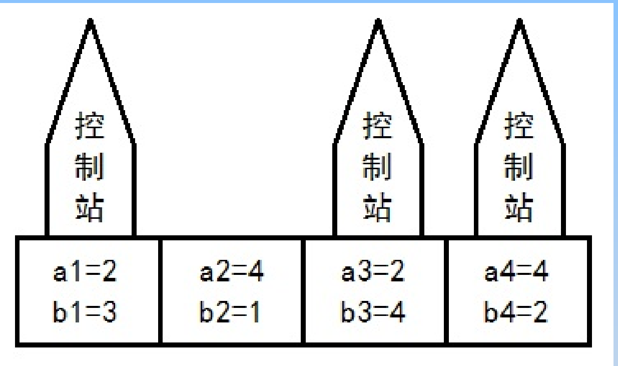
```
输入;
for(i=1;i<=n;i++)
{
  while(队首的状态不是斜率为2as_i的切点)删除队首元素;
  队首元素→ i;
  while(队尾元素与i不满足凸性质)删除队尾元素;
  把i放在队尾;
}
输出;
```

小P的牧场 NKOJ2706

【题目大意】

· 有n个牧场排成一条直线，第i个牧场建立控制站的代价为a_i，放养量为b_i。每个控制站只能控制他左边第一个控制站到自己之间的每个牧场，每个牧场被控制的代价为它的放养量乘控制它的牧场到它的距离。求最小代价。

· 例如右图，1号牧场被控制代价为0，2号牧场被控制代价为1，3号牧场代价为0，4号代价为0，加上修建控制站的代价，总代价为9。



【数据范围】

$1 \le n \le 10^6, 1 \le a_i, b_i \le 10^4$

又来谈谈你的看法

又来看看DP：

- 设$f[i]$表示前i个牧场被控制的最小代价
- $$f[i]=a_i+\min_{1\le j\le i-1} f[j]+\sum_{k=j+1}^{i} b_k\times(i-k)$$
$$=a_i+\min_{1\le j\le i-1} f[j]+i\times\sum_{k=j+1}^{i} b_k-\sum_{k=j+1}^{i} k\times b_k$$
- 分别对b_k和$k\times b_k$用前缀和优化，记前缀和分别为c_i和d_i，然后尝试分离i,j。
- $$f[i]=a_i+\min_{1\le j\le i-1} f[j]+i\times\left(c_i-c_j\right)-\left(d_i-d_j\right)$$
$$=a_i+i\times c_i-d_i+\min_{1\le j\le i-1} f[j]-i\times c_j+d_j$$
- 于是我们又遇到了喜闻乐见的<u>$i\times c_j$这样的交叉项</u>，无法分离
- 怎么破？当然是用斜率优化轻松破！

斜率优化一下！

- 假设$j < k$且$j \to i$比$k \to i$更优，得到等价表达式：
- $f[j] - i \times c_j + d_j < f[k] - i \times c_k + d_k$

- 把与i相关的放到一边，其他放到另一边
- $i \times (c_k - c_j) < (f[k] + d_k) - (f[j] + d_j)$

- 再令$e_j = f[j] + d_j$，得到
- $i < \frac{e_k - e_j}{c_k - c_j}$

- 如果把(c_j, e_j)看做j号点的坐标，把(c_k, e_k)看做k号点的坐标，我们又得到的了一个关于斜率的式子。
- 同理可以得到当$j > k$时的式子，画出图像可以知道，这道题只需要维护下凸包，就可以像上一道题一样进行状态转移，总的时间复杂度是$O(n)$的，可以通过本题。

斜率优化到底是什么呢？

- 回顾刚才解题的过程，归纳斜率优化的基本步骤

1. 写出普通的DP状态转移方程；
2. 通过一些换元技巧使方程只有i,j两个字母；
3. 尽量分离i,j成两个部分，但是发现存在形如$f(i)*g(j)$的东西不能分离；
4. 假设$j<k$且$j\to i$比$k\to i$优，得到一个不等式
5. 把不等式中与i相关的放在一边，其他放在另一边
6. 如果需要就再进行一次换元，然后除下去(注意正负号可能会影响不等号的方向)
7. 判断需要维护上凸包还是下凸包。
8. 如果$f(i)$是单调的，则可以用双端队列优化到$O(n)$；否则可以用栈+二分查找优化到$O(n\log n)$。

想一想，刚才的做法需要哪些前提？

1. 状态必须是$f[i]$这样的一维状态，转移必须使$j \to i$这样的一维转移，而且j的范围必须是$1 \le j \le \mathrm{range}(i)$。其中$\mathrm{range}(i)$是一个单调递增的函数。也就是所谓的$1D1D$动态规划。

2. 状态转移方程化简后必须有且只有一个形如$f(i) \times g(j)$的项。其他的都不行（例如$f(i)\ mod\ g(j)$，$f_1(i) \times g_1(j) + f_2(i) \times g_2(j)$，$\sin\big(f(i) \times g(j)\big) \ldots$）。

3. $g(j)$必须具有单调性，否则“除下去”那一步无法确定不等号方向是否改变。

4. 斜率优化把$O(n^2)$的DP优化到$O(n)$就是极限了，不可能有比$O(n)$更优的复杂度。如果需要更强的优化，请另请高明吧。

练习题

NKOJ 1918 锯木厂选址　　难度 ★

NKOJ 1919 玩具装箱　　难度 ★

NKOJ 2340 帮忙　　难度 ★

NKOJ 2845 序列分割　　难度 ★ ★

NKOJ 1539 土地购买　　难度 ★ ★

其实斜率优化挺也简单的，不是吗？

谢谢！