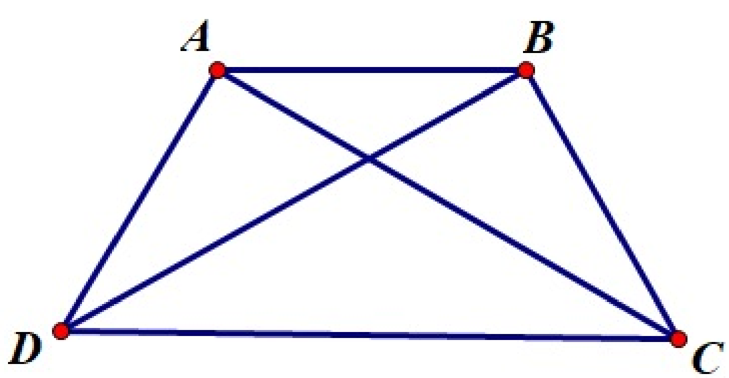
1 基于四边形不等式证明的决策单调性优化区间DP （例1）

1.1 什么是四边形不等式

1.1.1 数学角度出发

四边形不等式：如下图所示 $dis(A,D)+dis(B,C)\le dis(A,C)+dis(B,D)$ 四边形对边长度和≤对角线长度和



1. 可以借助三角形ABD和三角形ABC 显然证明。

1.1.2 信息学角度出发

考虑我们已经熟悉的区间DP，有常见转移方程$dp_{l,r} = \min(dp_{l,k} + dp_{k+1,r}) + w_{l,r}$， $k \in [l,r]$

这个模型一共有n^2 个状态，每次转移需要$O(n)$ 所以复杂度为$O(n^3)$

考虑优化转移过程复杂度，当$w_{l,r}$满足下列2个条件时，我们可以进一步优化。

1. 满足区间包含单调性，若区间$A[l_a, r_a]$ 被区间$B[l_b, r_b]$ 包含($l_b \le l_a \le r_a \le r_b$), 则 $w_{l_a,r_a} \le w_{l_b,r_b}$
2. 满足四边形不等式，若区间$A[l_a, r_a]$ 被区间$B[l_b, r_b]$ 包含($l_b \le l_a \le r_a \le r_b$), 则 $w_{l_b,r_a} + w_{l_a,r_b} \le w_{l_b,r_b} + w_{l_a,r_a}$

> 从DP角度看待问题，区间DP是一个2D/1D的动态规划。（约有n^2 个状态，每次转移需要$O(n)$ ）
>
> 1.2 中第2条，四边形不等式的满足是显然的，我们稍微画图即可得到答案：

1.2 四边形不等式如何优化区间DP(2D1D)

当$w_{i,j}$ 满足条件1 2 时，$dp_{i,j}$也满足四边形不等式。

假定我们已经记录了对于所有区间$dp_{i,j}$的最佳决策点 $best_{i,j}$ 那么

$$best_{l,r-1} \le best_{l,r} \le best_{l+1,r}$$

这个不等式告诉我们，最优决策点的二维矩阵是在行上，列上，单调不减的。这个叫决策单调性，关于决策单调性的具体说明，可见下方1.3 。

【优化区间DP】我们只需要动动小手，直接把区间DP的k值决策枚举范围从原来的$k \in [l,r]$ 变为 $k \in [best_{l,r-1}, best_{l+1,r}]$ 即可。

借助这种单调性，我们可以将复杂度压到$O(n^2)$

【code】

```
for (int i = 1; i <= n; ++i) m[i][i] = i; // 初始化边界决策点
for (int d = 2; d <= n; ++d){
    for (int l = 1, r = d; r <= n; ++l, ++ r){ //常规的区间DP
        dp[l][r] = INF;
        for (int k = m[l][r - 1]; k <= m[l + 1][r]; ++k) // 利用结论，缩小了枚举范围
            if (dp[l][k] + dp[k + 1][r] + w(l, r) < dp[l][r]){
                dp[l][r] = dp[l][k] + dp[k + 1][r] + w(l, r); // 更新dp数组
                m[l][r] = k; // 更新决策点
            }
    }
}
```

> 关键$best_{l,r-1} \le best_{l,r} \le best_{l+1,r}$不等式的证明：咕。（留下10个积分，期待在study空间看到大家的文字证明。只要写出来，都有5个积分，能当面找czc独立证明， 得10个积分。提示：https://oi-wiki.org/dp/opt/quadrangle/ ，蓝书）
>
> 复杂度 $O(n^2)$ 证明？10个积分。

3.1 什么是决策单调性

对于$dp_i = \min\limits_{0\le j<i}\{dp_j + w_{j,i}\}$ 这样的状态转移方程，记$best_i = j$，表示 dp_j 使 dp_i 最小（决策最优），若$best_i$ 在$i \in [1, N]$上单调不减（非严格递增）则说明dp_i 具有决策单调性。

当 $w_{j,i}$ 满足四边形不等式时，dp_i具有决策单调性。（证明：蓝书P330）

对于二维DP而言，其定义是类似的（1.2: 最优决策点的二维矩阵是在行上，列上，单调不减的。）

【重要观点】

本课最重要的内容，在于寻找DP过程中最优决策点的决策单调性，再根据单调性去优化DP。

四边形不等式只是在证明决策单调性！实际考场需要先猜测，再打表找规律确认。

2 四边形不等式对于其他DP模型的决策单调性的证明 （例2）

2.1 数学角度出发

1. 若 $f_{l,r}$ 和$g_{l,r}$ 满足前文1.2中提到的条件1 区间包含单调性，与条件2 四边形不等式。那么对于任意 $A, B \ge 0$ 来说， $Af_{l,r} + Bg_{l,r}$ 也满足四边形不等式，即$Af_{l,r} + Bg_{l,r}$也拥有**决策单调性**。
2. 若存在f_x, g_x使$w_{l,r} = f_r - g_l$则$w_{l,r}$符合四边形不等式。（若f_x, g_x单增，则$w_{l,r}$还满足区间包含单调性，$w_{l,r}$具有**决策单调性**。）
3. 若$h(x)$ 单增且下凸。$w_{l,r}$ 符合四边形不等式和区间包含单调性，则$h(w_{l,r})$ 也符合四边形不等式和区间包含单调性，$h(w_{l,r})$具有**决策单调性** 。
4. 若$h(x)$ 下凸。$w_{l,r}$ 符合四边形不等式和区间包含单调性，则$h(w_{l,r})$ 也符合四边形不等式。

2.2 信息学角度出发

实际上，如果发现DP复杂度是$O(n^3)$但是正好需要$O(n^2)$的复杂度（1e5数据范围）。这个时候就要大胆猜测是不是四边形不等式优化DP了。

真实的考场需要打表找规律验证猜想，如果是，就大胆用。证明的时间一般是不足的(因为证不出来）。

但是平时自己训练需要有一定的证明练习，不然无法形成数学/信息学直觉，对题目完成不利。

> 打表的对象是对决策点打表，记录每一段区间`[l,r]`对应的最佳决策点。

> 观察是否满足决策单调性。建议在时间允许范围内，把表打大一点，然后写代码随机抽点验证（表大了输出不好看）。

2.3 更多例子

https://www.cnblogs.com/Winniechen/p/9218864.html

3.3 三个案例

请结合2.1 和2.2 综合理解

【例1】

石子合并：其合并代价的计算过程由公式 $w_{l,r}=S_r-S_{l-1}$ ，可根据 3.1-2 得出符合四边形不等式满足单调性，可以使用四边形不等式证明决策单调性。

> S为前缀和数组。

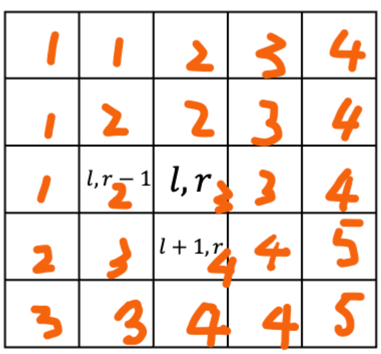
【例2】

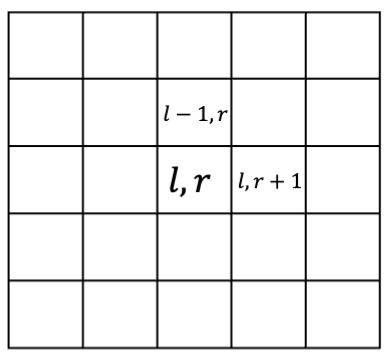
例如： $w_{l,r}=(r-l)^2$ ，可根据3.1-2，3.1-3分析得出，可以认为左侧公式为 $w_{l,r}=(t_{l,r})^2$ ，其中 $t_{l,r}=f_r-g_l$ 由于x^2是单增下凸函数，且f_x,g_x单增，则 $w_{l,r}=(r-l)^2$ 满足四边形不等式和区间包含单调性，可以使用四边形不等式证明决策单调性。

【例3】

实际上，除了传统的 $best_{l,r-1}\le best_{l,r}\le best_{l+1,r}$ 外，我们有可能遇到多种表现形式不同的不等式，例如 $best_{l-1,r}\le best_{l,r}\le best_{l,r+1}$

这其实是由于dp状态设计不一致导致的，本质上仍然是在反应**决策点在决策区间行列上的单调性**，如下图所示：





$best_{l,r-1}\le best_{l,r}\le best_{l+1,r}$　　　　$best_{l-1,r}\le best_{l,r}\le best_{l,r+1}$

> 图片解释，左图，表现了行列上的单调特点。
>
> 右图，行上单调递减（随着j增大而减小）代码第二层for 逆序即可处理。

3 决策单调性优化DP （核心内容）

3.1 二分+双端队列

当dp_i具有决策单调性时，我们可以通过对决策二分，将取min的O(n)优化到O(logn),整体复杂度优化到O(nlogn)

根据决策单调性，我们的best数组应该长这样（其中，j表示当前最优决策的点为j）

j_1	j_1	j_2	j_3	j_3	j_3	j_4	j_4	j_5	j_5	j_5

$j_1 < j_2 < j_3 < j_4 < j_5$

在决策过程中，假定我们突然发现某个位置上（例如i=6这个位置上）新求出了一个决策点位 pos 使得决策更优（意义：dp_{pos} 在$dp_{1\sim6}$的区间转移过程能取到更优秀的结果。）很有可能能dp_{pos} 还会是$dp_{1\sim7}$区间，$dp_{1\sim8}$区间，$dp_{1\sim9}$区间的最优决策点。

那么我们需要快速找出位置6（$i=6$）后面的某个位置i_2，要求dp_{1+i_2} 比 dp_i更优，使得这一片区域$best_i \sim best_{i_2}$均为pos，例如下图所示

j_1	j_1	j_2	j_3	j_3	pos	pos	pos	pos	pos	pos

$j_1 < j_2 < j_3 < j_4 < pos$

实际上，我们不能直接修改数组，所以我们可以将best数组映射到一组三元组(j,l,r)上，j表示决策点为dp_j, l,r表示当前区间 `[l,r]` 均为j

例如，上方第一个数组，我们可以抽象为：$(j_1,1,2),(j_2,3,3),(j_3,4,6),(j_4,7,8),(j_5,9,11)$

接下来，考虑如何转移到第二个数组

从队尾检查，发现$(j_4,7,8),(j_5,9,11)$ 比dp_{pos}的决策要差，直接删除。 然后在$(j_3,4,6)$ 中二分答案找到合适的位置为6，最后把$(j_3,4,5),(pos,6,11)$ 加入队尾即可。

因为队列中，没必要保存比$best_1 \sim best_{j-1}$ 还要小的决策点，所以直接排除这些不可能解后，我们的队首就是我们的最优决策。

【具体操作请见蓝书（下方）】

总而言之，对于每个 $i \in [1,N]$，我们执行以下操作：

1. 检查队头：设队头为 (j_0,l_0,r_0)，若 $r_0 = i-1$，删除队头。否则令 $l_0 = i$。
2. 取队头保存的 j 为最优决策，执行状态转移，计算出 $F[i]$。
3. 尝试插入新决策 i，步骤如下：
   (1) 取出队尾，记为 (j_t,l_t,r_t)。
   (2) 若对于 $F[l_t]$ 来说，i 是比 j_t 更优的决策，即 $F[i]+val(i,l_t) \le F[j_t]+val(j_t,l_t)$，记 $pos=l_t$，删除队尾，回到步骤(1)。
   (3) 若对于 $F[r_t]$ 来说，j_t 是比 i 更优的决策，即 $F[j_t]+val(j_t,r_t) \le F[i]+val(i,r_t)$，去往步骤(5)。
   (4) 否则，在 $[l_t,r_t]$ 上二分查找，求出位置 pos，在此之前决策 j_t 更优，在此之后决策 i 更优，去往步骤(5)。
   (5) 把三元组 (i,pos,N) 插入队尾。

请读者结合以下例题尝试实现上述过程。

例：诗人小G

小G是一个出色的诗人，经常作诗自娱自乐。但是，他一直被一 件事情所困扰，那就是诗的排版问题。

一首诗包含了若干个句子，对于一些连续的短句，可以将它们用 空格隔开并放在一行中，注意一行中可以放的句子数目是没有限 制的。小G给每首诗定义了一个行标准长度（行的长度为一行中 符号的总个数），他希望排版后每行的长度都和行标准长度相差 不远。

显然排版时，不应改变原有的句子顺序，并且小G不允许 把一个句子分在两行或者更多的行内。在满足上面两个条件的情 况下，小G对于排版中的每行定义了一个**不协调度**, 为**这行的实际 长度与行标准长度差值绝对值的P次方**，而**一个排版的不协调度 = 所有行不协调度的总和。**

小G最近又作了几首诗，现在请你对这首诗进行排版，使得排版 后的诗尽量协调（即不协调度尽量小），并把排版的结果告诉他。

【分析】

状态定义： f_i ，前i句排版的最小不协调度.

状态转移：$f_i = \min\limits_{0 \le j \le i} \{f_j + |sum_i - sum_j + (i-j-1) - L|^P\}$

$w_{i,j} = |sum_i - sum_j + (i-j-1) - L|^P$ 是否满足四边形不等式?

根据四边形不等式定义，想要证明其满足四边形不等式，

【方法1】

只需要证明$\forall j < i, w_{j,i+1} + w_{j+1,i} \ge w_{j,i} + w_{j+1,i+1}$

即,证明 $\forall j < i, w_{j+1,i} - w_{j+1,i+1} \ge w_{j,i} - w_{j,i+1}$

令，$u = (sum_i + i) - (sum_j + j) - (L+1), v = (sum_i + i) - (sum_{j+1} + j + 1) - (L+1)$

则有：$|v|^P - |v + a_{i+1} + 1|^P \ge |u|^P - |u + a_{i+1} + 1|^P$

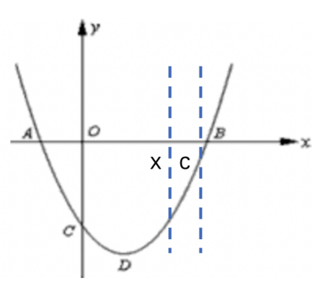
显然$u > v$ 我们令$u = v = x$,$a_{i+1} + 1 = c$

有 $|x|^P - |x+c|^P \ge |x|^P - |x+c|^P$，

结合$u > v$ 我们只需要证明$y = |x|^P - |x+c|^P$ 对于任意c单调递减即可。

证明方法1：严格分类讨论（$x < -c$,$-c \le x \le 0$,$x > 0$与P的奇偶性）+求导看函数单调性，

感性方法2：Case1 当$P = 1$时，$y = c$，符合。 Case2（下图）： 当$P \ne 1$时，x越大， x P与 $x + c$ P的差距就会越大，得证。



【方法2】

方向，尝试利用2.1 中的性质

【方法3】

尝试打表。找规律（普遍做法）

还可以顺便写个分段数据。暴力分用暴力做法，std分用std做法，尽可能保证你稳。（千万别写错暴力了。。。）

3.3 分治

【问题背景】

这是一个小技巧。

基于2D1D问题：$dp_{i,j} = \min\limits_{0<k\leq j}\{dp_{i-1,k} + w_{k,j}\}, i \in [1,n], j \in [1,m]$

当该dp转移方程中的$w_{k,j}$ 满足四边形不等式与区间包含单调性时，决策$best_{i,j}$具有单调性,即 $best_{i,j} \leq best_{i,j+1}$

【问题提出】

有些时候，我们不能在$O(logn)$ 或$O(1)$的时间复杂度内求出$w_{k,j}$

得益于决策的单调性（$best_{i,j} \leq best_{i,j+1}$） 可以考虑尝试使用整体二分来优化DP过程中的最优决策求解。

基于分治思想，我们需要对$best_{i,j}$区间二分，递归得到$best_{i,\frac{m}{2}}$, $best_{i,\frac{m}{4}}$ 这样，当想计算任意$best_{i,j}$的值时，复杂度最多 $O(logm)$

> 例题：FZUOJ 6225. 「CF321E」Ciel and Gondolas ，积分题。

4 广泛思考

本质上来讲，基于单调性思想进行DP过程优化的方法还有：

1. 斜率优化：每一个决策可以看做一个二维平面上的点，某两个决策的优劣性可以通过他们之间的斜率得出，并且随着决策的不断进行，我们所用来比较的斜率也是单调变化的，这样我们就可以把决策点做成一个凸包，用单调队列维护就可以了。
2. 分治：既然我们已经知道了决策时单调的，如果我们一个决定决策的复杂度不高，并且不依赖于其他的决策，那么我们可以分治进行。
   具体来说，我们每次暴力找出mid处的决策点，然后分治左右区间进行，这样就减少了每次决策的复杂度。（但是该模型泛化能力不足，如果决策之间相互依赖，就G）
3. 决策单调性：已经在本章提及。

得到决策单调性后，一般有两种方法解决问题：

1. 二分+双端队列：我们按照顺序进行dp，并维护每个位置**当前状态**下的最优转移位置。（蓝书上介绍的算法）yi些细节：在依次计算dp时，要把队首没用的弹出。在二分查找时，左端点要和i+1取max。要特判i不能更新任何位置的情况。
2. 3.3 分治：对区间分治，将最优决策点的转移范围缩小到logm范围内。

参考

https://zhuanlan.zhihu.com/p/398419302

https://oi-wiki.org/dp/opt/quadrangle/

https://www.cnblogs.com/resftlmuttmotw/p/12002741.html