极其简单的 DP 选讲

ljfcnyali
September 23, 2019

目录

一些 DP 技巧

[SDOI2011] 拦截导弹

有 n 个导弹，每个导弹有高度 h 和速度 v，拦截一个导弹的要求是高度和速度不能增加，求最多能拦截多少导弹

$n \leq 5 \times 10^4, h_i, v_i \leq 1000$

[SDOI2011] 拦截导弹

这道题 DP 方程特别简单，设 dp_i 表示拦截第 i 个导弹时最多可以拦截多少导弹，有方程 $dp_i = \max_{1\le j\le i-1}(dp_j + 1)$

[SDOI2011] 拦截导弹

这道题 DP 方程特别简单，设 dp_i 表示拦截第 i 个导弹时最多可以拦截多少导弹，有方程 $dp_i = \max_{1\le j\le i-1}(dp_j + 1)$

主要考虑优化状态转移，有一种很巧妙的方式是使用 CDQ 分治 (还可以用 KD-Tree?)

[SDOI2011] 拦截导弹

这道题 DP 方程特别简单，设 dp_i 表示拦截第 i 个导弹时最多可以拦截多少导弹，有方程 $dp_i = \max_{1 \le j \le i-1}(dp_j + 1)$

主要考虑优化状态转移，有一种很巧妙的方式是使用 CDQ 分治 (还可以用 KD-Tree?)

其实在 CDQ 分治里只需要改变 CDQ 分治的顺序，先递归计算 $l - Mid$，再计算左半对右半的更新，最后递归计算 $Mid + 1 - r$

[SDOI2011] 拦截导弹

这道题 DP 方程特别简单，设 dp_i 表示拦截第 i 个导弹时最多可以拦截多少导弹，有方程 $\mathrm{dp}_i = \max_{1 \le j \le i-1}(\mathrm{dp}_j + 1)$

主要考虑优化状态转移，有一种很巧妙的方式是使用 CDQ 分治 (还可以用 KD-Tree?)

其实在 CDQ 分治里只需要改变 CDQ 分治的顺序，先递归计算 $l - \mathrm{Mid}$，再计算左半对右半的更新，最后递归计算 $\mathrm{Mid} + 1 - r$

为什么是对的？

[SDOI2011] 拦截导弹

这道题 DP 方程特别简单，设 dp_i 表示拦截第 i 个导弹时最多可以拦截多少导弹，有方程 $dp_i = \max_{1\le j\le i-1}(dp_j + 1)$

主要考虑优化状态转移，有一种很巧妙的方式是使用 CDQ 分治 (还可以用 KD-Tree?)

其实在 CDQ 分治里只需要改变 CDQ 分治的顺序，先递归计算 $l - Mid$，再计算左半对右半的更新，最后递归计算 $Mid + 1 - r$

为什么是对的？

DP 转移需要满足转移必须有序，即不可以出现半成品，可以发现这种顺序相当于 CDQ 分治树的中序遍历

[SDOI2011] 拦截导弹

这道题 DP 方程特别简单，设 dp_i 表示拦截第 i 个导弹时最多可以拦截多少导弹，有方程 $dp_i = max_{1\le j\le i-1}(dp_j + 1)$

主要考虑优化状态转移，有一种很巧妙的方式是使用 CDQ 分治 (还可以用 KD-Tree?)

其实在 CDQ 分治里只需要改变 CDQ 分治的顺序，先递归计算 $l - Mid$，再计算左半对右半的更新，最后递归计算 $Mid + 1 - r$

为什么是对的？

DP 转移需要满足转移必须有序，即不可以出现半成品，可以发现这种顺序相当于 CDQ 分治树的中序遍历

这题其实还有一个第二问，但是与我们的标题不符就删去了

因为大家学习过斜率优化 DP，这里就只讨论一种基本解法

因为大家学习过斜率优化 DP，这里就只讨论一种基本解法

首先看一道例题

因为大家学习过斜率优化 DP，这里就只讨论一种基本解法

首先看一道例题

有一支由 n 个士兵组成的部队，要求分成连续的若干段，每一段的战斗力为 ax^2+bx+c，其中 x 是这一段士兵的战斗力和，求出最大的战斗力总和

斜率优化 [APIO2010] 特别行动队

因为大家学习过斜率优化 DP，这里就只讨论一种基本解法

首先看一道例题

有一支由 n 个士兵组成的部队，要求分成连续的若干段，每一段的战斗力为 $ax^2 + bx + c$，其中 x 是这一段士兵的战斗力和，求出最大的战斗力总和

$a < 0, n \leq 10^6$

设 sum_i 为前缀和，可以很简单的写出 DP 方程

设 sum_i 为前缀和，可以很简单的写出 DP 方程

$$\mathrm{dp}_i = \max_{0 \le j < i}\{\mathrm{dp}_j + a \times (\mathrm{sum}_i - \mathrm{sum}_j)^2 + b \times (\mathrm{sum}_i - \mathrm{sum}_j) + c\}$$

设 sum_i 为前缀和，可以很简单的写出 DP 方程

$dp_i = max_{0 \le j < i}\{dp_j + a \times (sum_i - sum_j)^2 + b \times (sum_i - sum_j) + c\}$

进行斜率优化推式子的基本步骤是假设由 j 转移比由 k 转移更优 ($k < j$，因为我们维护的是一个单调队列同时应弹出队首)，则有：

斜率优化 [APIO2010] 特别行动队

设 sum_i 为前缀和，可以很简单的写出 DP 方程

$$dp_i = \max_{0 \le j < i}\{dp_j + a \times (sum_i - sum_j)^2 + b \times (sum_i - sum_j) + c\}$$

进行斜率优化推式子的基本步骤是假设由 j 转移比由 k 转移更优 ($k < j$，因为我们维护的是一个单调队列同时应弹出队首)，则有：

$$dp_j - 2a sum_i sum_j + a sum_j^2 - b sum_j > dp_k - 2a sum_i sum_k + a sum_k^2 - b sum_k$$

设 sum_i 为前缀和，可以很简单的写出 DP 方程

$$dp_i = \max_{0\le j<i}\{dp_j + a\times(sum_i - sum_j)^2 + b\times(sum_i - sum_j) + c\}$$

进行斜率优化推式子的基本步骤是假设由 j 转移比由 k 转移更优 ($k<j$，因为我们维护的是一个单调队列同时应弹出队首)，则有：

$$dp_j - 2asum_i sum_j + asum_j^2 - bsum_j > dp_k - 2asum_i sum_k + asum_k^2 - bsum_k$$

接下来一步是把同时和 i,j 或 i,k 有关的项移到一边

设 sum_i 为前缀和，可以很简单的写出 DP 方程

$$dp_i = \max_{0 \le j < i}\{dp_j + a \times (sum_i - sum_j)^2 + b \times (sum_i - sum_j) + c\}$$

进行斜率优化推式子的基本步骤是假设由 j 转移比由 k 转移更优 ($k < j$，因为我们维护的是一个单调队列同时应弹出队首)，则有：

$$dp_j - 2asum_i sum_j + asum_j^2 - bsum_j > dp_k - 2asum_i sum_k + asum_k^2 - bsum_k$$

接下来一步是把同时和 i, j 或 i, k 有关的项移到一边

$$dp_j + asum_j^2 - bsum_j - dp_k - asum_k^2 + bsum_k > 2asum_i(sum_j - sum_k)$$

再将这一边的项中与 j, k 有关的除过去

再将这一边的项中与 j, k 有关的除过去

$$\frac{dp_j + a sum_j^2 - b sum_j - dp_k - a sum_k^2 + b sum_k}{sum_j - sum_k} > 2 a sum_i$$

再将这一边的项中与 j, k 有关的除过去

$$\frac{dp_j + asum_j^2 - bsum_j - dp_k - asum_k^2 + bsum_k}{sum_j - sum_k} > 2asum_i$$

注意到题目中保证 $a < 0$，则等式右边单调递减并一直为负，所以我们需要维护一个上凸壳

再将这一边的项中与 j, k 有关的除过去

$$\frac{dp_j + asum_j^2 - bsum_j - dp_k - asum_k^2 + bsum_k}{sum_j - sum_k} > 2asum_i$$

注意到题目中保证 $a < 0$，则等式右边单调递减并一直为负，所以我们需要维护一个上凸壳

弹队尾的时候应取小于号（上凸壳是小于号，下凸壳反之，等于号无所谓）

假设 DP 式为 $f_{i,j} = \min_{k=i}^{j-1}\{f_{i,k} + f_{k+1,j}\} + w_{i,j}$ 时，若满足四边形不等式时可以进行优化：

四边形不等式优化

假设 DP 式为 $f_{i,j} = \min_{k=i}^{j-1}\{f_{i,k} + f_{k+1,j}\} + w_{i,j}$ 时，若满足四边形不等式时可以进行优化：

四边形不等式：对于 $a \le b \le c \le d$ 均有 $w_{a,c} + w_{b,d} \le w_{a,b} + w_{c,d}$，若 $w_{i,j}$ 满足四边形不等式，则 $f_{i,j}$ 也满足四边形不等式

假设 DP 式为 $f_{i,j} = \min_{k=i}^{j-1}\{f_{i,k} + f_{k+1,j}\} + w_{i,j}$ 时，若满足四边形不等式时可以进行优化：

四边形不等式：对于 $a \le b \le c \le d$ 均有 $w_{a,c} + w_{b,d} \le w_{a,b} + w_{c,d}$，若 $w_{i,j}$ 满足四边形不等式，则 $f_{i,j}$ 也满足四边形不等式

假设 $f_{i,j}$ 是由 $s_{i,j}$ 转移得到 (最优决策点)，那么 $f_{i,j}$ 的最优决策点当且仅当在 $[s_{i,j-1}, s_{i+1,j}]$ 范围内

假设 DP 式为 $f_{i,j} = \min_{k=i}^{j-1}\{f_{i,k} + f_{k+1,j}\} + w_{i,j}$ 时，若满足四边形不等式时可以进行优化：

四边形不等式：对于 $a \le b \le c \le d$ 均有 $w_{a,c} + w_{b,d} \le w_{a,b} + w_{c,d}$，若 $w_{i,j}$ 满足四边形不等式，则 $f_{i,j}$ 也满足四边形不等式

假设 $f_{i,j}$ 是由 $s_{i,j}$ 转移得到 (最优决策点)，那么 $f_{i,j}$ 的最优决策点当且仅当在 $[s_{i,j-1}, s_{i+1,j}]$ 范围内

可以证明时间复杂度是 $O(n^2)$

四边形不等式优化 [IOI2000] 邮局

有 V 个村庄，给定村庄坐标，需要建立 P 个邮局，使得村庄到其最近的邮局的距离和最小

四边形不等式优化 [IOI2000] 邮局

有 V 个村庄，给定村庄坐标，需要建立 P 个邮局，使得村庄到其最近的邮局的距离和最小

$\mathsf{P} \leq 300, \mathsf{V} \leq 3000$

朴素做法很好想，设 $f_{i,j}$ 表示前 i 个村庄建立 j 个邮局的最小距离和，那么有 DP 方程

朴素做法很好想，设 $f_{i,j}$ 表示前 i 个村庄建立 j 个邮局的最小距离和，那么有 DP 方程

$$f_{i,j} = \min_{0 \le k < i}\{f_{k,j-1} + dis(k+1, i)\}$$

朴素做法很好想，设 $f_{i,j}$ 表示前 i 个村庄建立 j 个邮局的最小距离和，那么有 DP 方程

$$f_{i,j} = \min_{0\le k<i}\{f_{k,j-1} + dis(k+1, i)\}$$

其中 $dis(l, r)$ 表示在村庄 $[l, r]$ 中建立一座邮局的最小距离和，众所周知建立在中位数最优秀，可以进行预处理

朴素做法很好想，设 $f_{i,j}$ 表示前 i 个村庄建立 j 个邮局的最小距离和，那么有 DP 方程

$$f_{i,j} = \min_{0 \le k < i}\{f_{k,j-1} + dis(k+1, i)\}$$

其中 $dis(l, r)$ 表示在村庄 $[l, r]$ 中建立一座邮局的最小距离和，众所周知建立在中位数最优秀，可以进行预处理

发现 $dis_{i,j}$ 和 $f_{i,j}$ 均满足四边形不等式，可以直接优化，注意 i 需要倒序枚举

四边形不等式优化 [IOI2000] 邮局

朴素做法很好想，设 $f_{i,j}$ 表示前 i 个村庄建立 j 个邮局的最小距离和，那么有 DP 方程

$f_{i,j} = \min_{0 \le k < i}\{f_{k,j-1} + dis(k+1, i)\}$

其中 $dis(l, r)$ 表示在村庄 $[l, r]$ 中建立一座邮局的最小距离和，众所周知建立在中位数最优秀，可以进行预处理

发现 $dis_{i,j}$ 和 $f_{i,j}$ 均满足四边形不等式，可以直接优化，注意 i 需要倒序枚举

这题还有一种非常巧妙的办法.... 去看洛谷题解吧

四边形不等式优化决策单调性

假设 DP 式为 $f_i = \min_{j=1}^{i-1}\{f_j + w(i,j)\}$，并且 $w(i,j)$ 满足四边形不等式，则可以进行优化

四边形不等式优化决策单调性

假设 DP 式为 $f_i = \min_{j=1}^{i-1}\{f_j + w(i,j)\}$，并且 $w(i,j)$ 满足四边形不等式，则可以进行优化

这种式子经过证明最优决策点会单调递增，所以可以使用分治的方式得到答案

四边形不等式优化决策单调性

假设 DP 式为 $f_i = \min_{j=1}^{i-1}\{f_j + w(i,j)\}$，并且 $w(i,j)$ 满足四边形不等式，则可以进行优化

这种式子经过证明最优决策点会单调递增，所以可以使用分治的方式得到答案

分治步骤大概是暴力枚举 Mid 的最优决策点，再分为 l, Mid 和 $Mid+1, r$ 求解，这样会不断缩小暴力求解最优决策点的范围，可以证明时间复杂度为 $O(n\log_2 n)$

四边形不等式优化 [POI2011]Lightning Conductor

给定 a，求解所有整数 $p_i = \max_{j=1}^{n}\{a_j + \sqrt{|i-j|} - a_i\}$

给定 a，求解所有整数 $p_i = \max_{j=1}^{n}\{a_j + \sqrt{|i-j|} - a_i\}$

$n \le 5 \times 10^5, a_i \le 10^9$

四边形不等式优化 [POI2011]Lightning Conductor

给定 a，求解所有整数 $p_i = max_{j=1}^{n}\{a_j + \sqrt{|i-j|} - a_i\}$

$n \le 5 \times 10^5, a_i \le 10^9$

为了方便大家思考，给出一个输入输出和每个点的最优决策点

四边形不等式优化 [POI2011]Lightning Conductor

给定 a，求解所有整数 $p_i = \max_{j=1}^{n}\{a_j + \sqrt{|i-j|} - a_i\}$

$n \le 5 \times 10^5, a_i \le 10^9$

为了方便大家思考，给出一个输入输出和每个点的最优决策点

8 2 3 2 4 2 3 2 4，第一个数为 n，接下来 n 个数表示 a_i

四边形不等式优化 [POI2011]Lightning Conductor

给定 a，求解所有整数 $p_i = \max_{j=1}^{n}\{a_j + \sqrt{|i-j|} - a_i\}$

$n \le 5 \times 10^5, a_i \le 10^9$

为了方便大家思考，给出一个输入输出和每个点的最优决策点

8 2 3 2 4 2 3 2 4，第一个数为 n，接下来 n 个数表示 a_i

5 4 5 2 4 3 4 3，表示 p_i

四边形不等式优化 [POI2011]Lightning Conductor

给定 a，求解所有整数 $p_i = \max_{j=1}^{n}\{a_j + \sqrt{|i-j|} - a_i\}$

$n \leq 5 \times 10^5, a_i \leq 10^9$

为了方便大家思考，给出一个输入输出和每个点的最优决策点

8 2 3 2 4 2 3 2 4，第一个数为 n，接下来 n 个数表示 a_i

5 4 5 2 4 3 4 3，表示 p_i

8 8 8 8 8 4 4 4，表示每个点的最优决策点

四边形不等式优化 [POI2011]Lightning Conductor

给定 a，求解所有整数 $p_i = \max_{j=1}^{n}\{a_j + \sqrt{|i-j|} - a_i\}$

$n \le 5 \times 10^5, a_i \le 10^9$

为了方便大家思考，给出一个输入输出和每个点的最优决策点

8 2 3 2 4 2 3 2 4，第一个数为 n，接下来 n 个数表示 a_i

5 4 5 2 4 3 4 3，表示 p_i

8 8 8 8 8 4 4 4，表示每个点的最优决策点

0 0 2 2 4 4 4 4，表示只考虑 $j < i$ 的情况的最优决策点

四边形不等式优化 [POI2011]Lightning Conductor

给定 a，求解所有整数 $p_i = \max_{j=1}^{n}\{a_j + \sqrt{|i-j|} - a_i\}$

$n \leq 5 \times 10^5, a_i \leq 10^9$

为了方便大家思考，给出一个输入输出和每个点的最优决策点

8 2 3 2 4 2 3 2 4，第一个数为 n，接下来 n 个数表示 a_i

5 4 5 2 4 3 4 3，表示 p_i

8 8 8 8 8 4 4 4，表示每个点的最优决策点

0 0 2 2 4 4 4 4，表示只考虑 $j < i$ 的情况的最优决策点

8 8 8 8 8 8 8 4，表示值考虑 $j \geq i$ 的情况的最优决策点

四边形不等式优化 [POI2011]Lightning Conductor

观察到这个绝对值不太方便，把绝对值拆开分两种情况

四边形不等式优化 [POI2011]Lightning Conductor

观察到这个绝对值不太方便，把绝对值拆开分两种情况

打表会发现拆开后的 p_i 满足决策单调性，其中 $j < i$ 的情况下 p_i 单调递增，反之单调递减

观察到这个绝对值不太方便，把绝对值拆开分两种情况

打表会发现拆开后的 p_i 满足决策单调性，其中 $j < i$ 的情况下 p_i 单调递增，反之单调递减

直接照上面的分治方式求解即可

四边形不等式优化 [POI2011]Lightning Conductor

观察到这个绝对值不太方便，把绝对值拆开分两种情况

打表会发现拆开后的 p_i 满足决策单调性，其中 $j < i$ 的情况下 p_i 单调递增，反之单调递减

直接照上面的分治方式求解即可

总结一下这些优化决策区间的题有两种情况，有决策上下界或单调性，具体最好打表来判断，但解法都是类似的

四边形不等式优化 [CF321E]Ciel and Gondolas [BZOJ5311] 贞鱼

将长度为 n 的序列分为 k 段，当 i, j 在同一段中，会产生 $a_{i,j}$ 的代价，求最小总代价

四边形不等式优化 [CF321E]Ciel and Gondolas [BZOJ5311] 贞鱼

将长度为 n 的序列分为 k 段，当 i, j 在同一段中，会产生 $a_{i,j}$ 的代价，求最小总代价

$n \le 4000, k \le 800$

设 $dp_{i,j}$ 表示前 i 个数分为 j 段的最小总代价，有方程

设 $dp_{i,j}$ 表示前 i 个数分为 j 段的最小总代价，有方程

$$dp_{i,j} = \max_{0 \le k < i}\{dp_{k,j-1} + sum(k+1, i)\}$$

设 $dp_{i,j}$ 表示前 i 个数分为 j 段的最小总代价，有方程

$$dp_{i,j} = max_{0 \le k < i}\{dp_{k,j-1} + sum(k+1, i)\}$$

$sum_{i,j}$ 可以用前缀和直接求出，同时可以发现在 j 固定的情况下，$dp_{i,j}$ 满足决策单调性，其中

设 $dp_{i,j}$ 表示前 i 个数分为 j 段的最小总代价，有方程

$$dp_{i,j} = \max_{0 \le k < i}\{dp_{k,j-1} + sum(k+1, i)\}$$

$sum_{i,j}$ 可以用前缀和直接求出，同时可以发现在 j 固定的情况下，$dp_{i,j}$ 满足决策单调性，其中

可以使用上面的分治方式，时间复杂度 $O(nk\log_2 n)$，CF 可以过，但 BZOJ 上 TLE 了

设 $dp_{i,j}$ 表示前 i 个数分为 j 段的最小总代价，有方程

$$dp_{i,j} = max_{0 \le k < i}\{dp_{k,j-1} + sum(k+1, i)\}$$

$sum_{i,j}$ 可以用前缀和直接求出，同时可以发现在 j 固定的情况下，$dp_{i,j}$ 满足决策单调性，其中

可以使用上面的分治方式，时间复杂度 $O(nklog_2n)$，CF 可以过，但 BZOJ 上 TLE 了

有一种优化叫 wqs 优化，可以把时间优化成 $O(nlog_2klog_2n)$，但是我不会....

插头 DP

插头 DP [HNOI2004] 邮递员

给定 $n \times m$ 的矩阵，求经过所有点的闭合回路个数

插头 DP [HNOI2004] 邮递员

给定 $n \times m$ 的矩阵，求经过所有点的闭合回路个数

$m \leq 10, n \leq 20$

插头 DP [HNOI2004] 邮递员

给定 $n \times m$ 的矩阵，求经过所有点的闭合回路个数

$m \leq 10, n \leq 20$

插头 DP 模板题，答案很大注意开 __int128

插头 DP [UVA10572]Black&White

一个棋盘，已有部分被染色为黑或白，给所有未染色的格子染上黑或白，使得所有黑格子在同一个连通块，所有白格子在同一连通块。并且不存在一个 2×2 的方格中的所有格子颜色相同，求染色方案数并输出一种合法方案

插头 DP [UVA10572]Black&White

一个棋盘，已有部分被染色为黑或白，给所有未染色的格子染上黑或白，使得所有黑格子在同一个连通块，所有白格子在同一连通块。并且不存在一个 2×2 的方格中的所有格子颜色相同，求染色方案数并输出一种合法方案

$\mathsf{n}, \mathsf{m} \leq 8, \mathsf{T} \leq 10$

插头 DP [UVA10572]Black&White

首先考虑输出一种合法方案，只需要记录一下任意一个得到答案的路径即可

插头 DP [UVA10572]Black&White

首先考虑输出一种合法方案，只需要记录一下任意一个得到答案的路径即可

因为不能有 2×2 的连通块，所以对于每一个点要记录一下其左上方的点，另外维护的状态应该包含颜色和连通性

插头 DP [UVA10572]Black&White

首先考虑输出一种合法方案，只需要记录一下任意一个得到答案的路径即可

因为不能有 2×2 的连通块，所以对于每一个点要记录一下其左上方的点，另外维护的状态应该包含颜色和连通性

因为这道题不需要插头，所以最小表示法使用记录为第几个连通块，每一次状态转移中连通块个数最多为 m 个

首先考虑输出一种合法方案，只需要记录一下任意一个得到答案的路径即可

因为不能有 2×2 的连通块，所以对于每一个点要记录一下其左上方的点，另外维护的状态应该包含颜色和连通性

因为这道题不需要插头，所以最小表示法使用记录为第几个连通块，每一次状态转移中连通块个数最多为 m 个

分类讨论即可 (我没调处来... 弃疗了)

动态 DP

动态 DP

看 oi-wiki.org，里面的是我写的....

练习题

删去了几道题，放到后面可以当做练习题做做

练习题

删去了几道题，放到后面可以当做练习题做做

[NOI2007] 货币兑换：这道题是 CDQ 分治 + 斜率优化，算是一开始讲的两个知识点的综合应用

练习题

删去了几道题，放到后面可以当做练习题做做

[NOI2007] 货币兑换：这道题是 CDQ 分治 + 斜率优化，算是一开始讲的两个知识点的综合应用

[NOI2014] 购票：这道题是树上的斜率优化，因为斜率的等式右边不满足单调性需要二分（或三分）维护，另外有一些距离限制可以用树剖或点分治解决

练习题

删去了几道题，放到后面可以当做练习题做做

[NOI2007] 货币兑换：这道题是 CDQ 分治 + 斜率优化，算是一开始讲的两个知识点的综合应用

[NOI2014] 购票：这道题是树上的斜率优化，因为斜率的等式右边不满足单调性需要二分（或三分）维护，另外有一些距离限制可以用树剖或点分治解决

[NOI2009] 诗人小 G：这题是四边形不等式的练习题，但是需要二分来维护决策单调性，很有意思

删去了几道题，放到后面可以当做练习题做做

[NOI2007] 货币兑换：这道题是 CDQ 分治 + 斜率优化，算是一开始讲的两个知识点的综合应用

[NOI2014] 购票：这道题是树上的斜率优化，因为斜率的等式右边不满足单调性需要二分（或三分）维护，另外有一些距离限制可以用树剖或点分治解决

[NOI2009] 诗人小 G：这题是四边形不等式的练习题，但是需要二分来维护决策单调性，很有意思

[GDKOI2016]Map：跟上面那道插头 DP 有点类似，也是维护连通性，可以用来练练手

Thanks