Solution

洛水 · 锦依卫

2019 年 8 月 10 日

写在前面

题目难度递减，总是还是按难度排序的对吧。

1 T1 sow

先考虑没有修改的情况。

设 f_i 为已经处理到 i 时，最大的收获，那么 i 可以作为一个选择了的区间的结尾，也可以不选。再设 $F(x)=\frac{x(x+1)}{2}$ ，前缀和为 s_i ，那 dp 转移方程为：

$$f_i=max(f_{i-1},Max\{f_j+F(i-j)-s_i+s_j\})$$

很显然的，我们发现可以斜率优化。

若有 $j<k$ 且从 j 转移比 k 更优，则有：

$$f_j+\frac{(i-j)*(i-j+1)}{2}-s_i+s_j>f_k+\frac{(i-k)*(i-k+1)}{2}-s_i+s_k$$

$$f_j-f_k+s_j-s_k>\frac{2(j-k)*i+F(k-1)-F(j-1)}{2}$$

$$\frac{f_j+s_j+F(j-1)-f_k-s_k-F(k-1)}{j-k}<i$$

由于 i 递增，所以满足决策单调性。

所以可以维护一个斜率单调递减或者说上凸壳的栈，每次取决策点时比较栈顶与第二个元素并决定弹出。取栈顶转移即可。

那么有修改呢？其实也很简单。

对于修改点 x ，其收获的计算只会基于两种情况：

一、方案中没有选择点 x 。那么我们只需要计算出未修改时的 dp 前缀值 f 与后缀值 g ，没有选择 x 的答案就是 $f_{x-1}+g_{x+1}$ 。

二、方案中选择了点 x 。

我们考虑一件事情，那就是：若当前必须选择点 x ，那么无论 x 是否被修改，最大收获方案唯一。

则我们可以考虑求出 $must_i$ 表示若必须选点 i ，未修改时整个序列的最大收获。那么选择了点 x 的答案就是 $must_i-w_x+y$ 。我们只需要将选与没选的答案取 max 即可。

那么问题在于：如何求 $must$ 数组？

接下来操作比较社会高级了：

如果暴力求 $must$ 数组要怎么求呢？对于 $must_i$ ，我可以枚举所有覆盖了 i 的区间 $[l,r]$ ，在所有 $f_{l-1}+g_{r+1}+F(r-l+1)$ 中取 MAX 。但这样做复杂度会很劣。

考虑分治，没错就是分治。

假设当前分治区间为 $[l,r]$ ，我们只需要考虑枚举覆盖了 mid 的区间就好了。

考虑枚举 $R\in[mid,r]$ 作为区间右端点，左端点位于区间 $[l,mid]$ 的最大收获情况。

那么我们只需要设 G_R 为以 R 为右端点时最大收获的情况，则有如下式子：

$$G_R=Max\{f_{L-1}-s_R+s_{L-1}+F(R-L+1)\}+g_{R+1}$$

然后我们发现这个也可以斜率优化哦……

那么对于 G 数组取后缀最大值,因为 G_R 可以同时用来更新 $\forall i\in[mid,r],must_i$ 。

对于 $\forall i\in[l,mid],must_i$ ，我们也可以同理更新。

然后递归下去就好了！

2 T2 canal

首先我们发现题目里的条件很 $interesting$: 种子都只会被分为两列。这个条件有什么用呢？二分图？当然不是。毕竟这是 $NOIP$ 模拟题 $==$。

2.1 推论一

可以考虑手画一下各种情况，我们就能得到一个推论：

最后的最小生成树中绝对不包含交叉的两条线段。

为什么呢？假设左侧有点 A 与点 B ，右侧有点 C 与点 D ，且 AD 与 BC 在 MST 内，且两线段交叉，请手画一下这四个点，那我们考虑断开这两条边的情况：

首先，由于是这是一棵 MST ，所以我们断开 AD 与 BC 后会分成三个联通块。

先断开 AD ，设此时 B,C 与 A 在同一联通块内（与 D 同联通块同理)。

再断开 BC ，设此时 A 与 B 在同一联通块内（与 C 同联通块同理)。

那么此时，连接 AC 与 BD 可以使树再次联通，且由于 AB 与 CD 平行，$AC+BD$ 必定小于 $AD+BC$ 这两条交叉线段。

2.2 推论二

若有点 $A(x1,y1),B(x1,y2),C(x2,y3)$ ，且 $y1<y2<y3$ 。

若在 MST 中，BC 之间有连线，那么由于线段不能相交，所以在 B 上方的点与 C 下方的点都不会有连线，且 C 上方与 B 下方也不会有连线。

那么考虑 A 必定在 BC 下方的一个联通块内，且由于线段不相交，BC 所在的联通块若想与 A 所在的联通块合并，必须在两个联通块之间连线。

分别考虑连接 AC 与 AB 的情况，由于联通性的扩展相同，所以我们只需考虑边长问题。

那么因为 $y1<y2<y3$ ，则 $\triangle ABC$ 为钝角三角形，那么 AC 为最长边。所以连 AB 比 AC 更优。

接着，考虑若 BC 之间没有边，则我们需要连接 A,B,C 所在的三个联通块，$BC+AB$ 的连接决策必然比 $AB+AC$ 或是 $AC+BC$ 更优。

简而言之，对于这样的三个点 A,B,C ，边 AC 必定不会存在于 MST 中。

那么有了这两个推论，可以得到，对于每一个点，它最多连出去四条边，一条连上面，一条连下面，连到对面的边也最多只有两条，大致可以理解为 y 坐标在当前点上下的最近的两个点，那么我们的边数就可以减少到 $2(n+m)$ 级别，这个时候做一下最小生成树可以拿到 $70\sim 100$ 分的好成绩，具体取决于脸常数。

当然，既然边长得都这么好看了，显然可以直接 DP 的呀。

设 $f[i][j][0/1]$ 表示左边第 i 个点与右边第 j 个点以下的点已经处理完了，且点 i,j 是否连通的最小长度。

则 $f[i][j]$ 可以从 $f[i-1][j]$ 与 $f[i][j-1]$ 转移过来。由于 MST 中的连边必须满足若两侧的点之间连边，必须不存在中间点能构成钝角三角形，那么可供状态表示的 i,j 对数不超过 n 对。

设 $s[i]$ 为左侧点的坐标，可以得到 $f[i][j]$ 从 $f[i-1][j]$ 转移的方程 (从 $f[i][j-1]$ 同理)：

$$f[i][j][1] = min(f[i-1][j][1]+min(s[i]-s[i-1], dis(i,j)), f[i-1][j][0]+s[i]-s[i-1]+dis(i,j))$$

$$f[i][j][0] = min(f[i-1][j][1], f[i-1][j][0] + s[i] - s[i-1])$$

同时我们发现这个状态是可以滚的⋯⋯所以空间使用很小⋯⋯设一个 $f[2][2][2]$ 的 dp 数组即可。

3 T3 oitree

原题 + 水题 + 签到题，送 100 分，谁都没资格再说我毒瘤了。

就算没有 100 ，这么多简单好拿的暴力分也很良心。

而且还有好几个人切过，预计全场 A 。

很简单啊，虽然当所有叶节点都多出一条到达根节点的路径时会导致两点之间路径不唯一，但是也不唯一得很有限。

在这种情况下，两点之间还是只有大致上两种走法。

一，直接走树上路径。

二，先走树上路径，然后在半路某个点处先走到根节点，再走到终点。

第一种路径的长度很好求，那第二种我们如何求最优的呢？看着很多，但实际上还是一个最优值的套路。

我们先处理出所有点到根节点的最短路，用 $dis[i]$ 表示从点 i 到点 1 的最短路，反之亦然。接着设 $deep[i]$ 为 i 到根节点的前缀路径长度。

那么对于 $u->v$ 这条树上路径上的一点 i ，如果我们选择从点 i 到根节点再到终点，则我们的路径长度也很好求。

先设 lca 为 u,v 的最近公共祖先，如果 i 在 $u->lca$ 这条路径上，那么路径长度为：

$$deep[u] - deep[i] + dis[i] + dis[v]$$

而如果 i 在 $lca->v$ 这条路径上，那么路径长度为：

$$deep[u]-deep[lca]+deep[i]-deep[lca]+dis[i]+dis[v]$$

对于第二种路径，我们要求的其实就是：

$$Min_{i\in u->lca}(dis[i]-deep[i])+dis[v]+deep[u]$$

$$Min_{i\in lca->v}(deep[i]+dis[i])+deep[u]-2*deep[lca]+dis[v]$$

辣么，不难发现，我们只需要将路径拆成两个部分，对于 $u->lca$ 这部分上的点，取 $dis[i]-lj[i]$ 的最小值处并计算，对于 $lca->v$ 这部分上的点，取 $lj[i]+dis[i]$ 的最小值处并计算，然后对于这两部分的最优值取 min 即可。

然后将第二种路径与第一种路径相比较取 min 就是询问的答案。

对于维护路径两部分上的不同最小值，我们用~~树链剖分~~倍增来维护两个树上 ST 表即可。