Solution

ExfJoe

福建省长乐第一中学

March 10, 2017

Outline

圆圈游戏

圆圈游戏

- 圆之间的包含关系可以看做一个森林

圆圈游戏

- 圆之间的包含关系可以看做一个森林
- 添加一个圆心为 $(0,0)$ 半径为无穷大的圆，则包含关系可建为一棵有根树

圆圈游戏

- 圆之间的包含关系可以看做一个森林
- 添加一个圆心为 $(0,0)$ 半径为无穷大的圆，则包含关系可建为一棵有根树
- $dp_{i,0/1}$ 表示点 i 不选/选时可得的最大价值，树形 DP 即可

圆圈游戏

圆圈游戏

- 圆之间的包含关系用扫描线处理出来

圆圈游戏

- 圆之间的包含关系用扫描线处理出来
- 在 $x_i - r_i$ 时插入一个圆，在 $x_i + r_i$ 时删除一个圆

圆圈游戏

- 圆之间的包含关系用扫描线处理出来
- 在 $x_i - r_i$ 时插入一个圆，在 $x_i + r_i$ 时删除一个圆
- 插入删除圆时把圆拆成两个半圆操作

圆圈游戏

- 圆之间的包含关系用扫描线处理出来
- 在 $x_i - r_i$ 时插入一个圆，在 $x_i + r_i$ 时删除一个圆
- 插入删除圆时把圆拆成两个半圆操作
- 圆的相对顺序不会变，因此插入时查找当前下半圆的前驱 (先查找再插入)，若是一个上半圆，则它们是父子关系，若为下半圆，则为兄弟关系

圆圈游戏

- 圆之间的包含关系用扫描线处理出来
- 在 $x_i - r_i$ 时插入一个圆，在 $x_i + r_i$ 时删除一个圆
- 插入删除圆时把圆拆成两个半圆操作
- 圆的相对顺序不会变，因此插入时查找当前下半圆的前驱 (先查找再插入)，若是一个上半圆，则它们是父子关系，若为下半圆，则为兄弟关系
- 平衡树维护，时间复杂度 $O(n\log n)$

Outline

划分序列

划分序列

- 显然可以二分答案 ans

划分序列

- 显然可以二分答案 ans
- $A_i \geq 0$：求出最小划分段数 x，若 $x \leq K$ 则当前答案可行

划分序列

- 显然可以二分答案 ans
- $A_i \geq 0$：求出最小划分段数 x，若 $x \leq K$ 则当前答案可行
- $A_i \leq 0$：求出最大划分段数 x，若 $x \geq K$ 则当前答案可行

划分序列

- 显然可以二分答案 ans
- $A_i \geq 0$：求出最小划分段数 x，若 $x \leq K$ 则当前答案可行
- $A_i \leq 0$：求出最大划分段数 x，若 $x \geq K$ 则当前答案可行
- 满分做法：根据上面两种情况以及暴力对拍，我们可以发现：

划分序列

- 显然可以二分答案 ans
- $A_i \geq 0$：求出最小划分段数 x，若 $x \leq K$ 则当前答案可行
- $A_i \leq 0$：求出最大划分段数 x，若 $x \geq K$ 则当前答案可行
- 满分做法：根据上面两种情况以及暴力对拍，我们可以发现：
- 若最小划分段数为 L，最大划分段数为 R，当前答案可行当且仅当 $L \leq K \leq R$

划分序列

- 显然可以二分答案 ans
- $A_i \geq 0$：求出最小划分段数 x，若 $x \leq K$ 则当前答案可行
- $A_i \leq 0$：求出最大划分段数 x，若 $x \geq K$ 则当前答案可行
- 满分做法：根据上面两种情况以及暴力对拍，我们可以发现：
- 若最小划分段数为 L，最大划分段数为 R，当前答案可行当且仅当 $L \leq K \leq R$
- L, R 均可以用 DP 求出

划分序列

- 显然可以二分答案 ans
- $A_i \geq 0$：求出最小划分段数 x，若 $x \leq K$ 则当前答案可行
- $A_i \leq 0$：求出最大划分段数 x，若 $x \geq K$ 则当前答案可行
- 满分做法：根据上面两种情况以及暴力对拍，我们可以发现：
- 若最小划分段数为 L，最大划分段数为 R，当前答案可行当且仅当 $L \leq K \leq R$
- L, R 均可以用 DP 求出
- 时间复杂度 $O(n \log^2 n)$

Outline

1 圆圈游戏

2 划分序列

3 生成树求和

生成树求和

生成树求和

- 容易想到矩阵树定理，但矩阵树定理只能求边上权值积的和，而不能求和的和

生成树求和

- 容易想到矩阵树定理，但矩阵树定理只能求边上权值积的和，而不能求和的和
- 考虑按三进制逐位处理，若一条边边权为 i，则我们将其视为 x^i

生成树求和

- 容易想到矩阵树定理，但矩阵树定理只能求边上权值积的和，而不能求和的和
- 考虑按三进制逐位处理，若一条边边权为 i，则我们将其视为 x^i
- 考虑这种情况下跑矩阵树，则得出的是一个次数不超过 $2n$ 的多项式，x^i 的系数 A_i 即当前位边权和为 i 的方案数

生成树求和

- 容易想到矩阵树定理，但矩阵树定理只能求边上权值积的和，而不能求和的和
- 考虑按三进制逐位处理，若一条边边权为 i，则我们将其视为 x^i
- 考虑这种情况下跑矩阵树，则得出的是一个次数不超过 $2n$ 的多项式，x^i 的系数 A_i 即当前位边权和为 i 的方案数
- $A_i \times (i \bmod 3)$ 即它的贡献

生成树求和

- 容易想到矩阵树定理，但矩阵树定理只能求边上权值积的和，而不能求和的和
- 考虑按三进制逐位处理，若一条边边权为 i，则我们将其视为 x^i
- 考虑这种情况下跑矩阵树，则得出的是一个次数不超过 $2n$ 的多项式，x^i 的系数 A_i 即当前位边权和为 i 的方案数
- $A_i \times (i \bmod 3)$ 即它的贡献
- 但直接用多项式无法方便求解行列式

生成树求和

- 容易想到矩阵树定理，但矩阵树定理只能求边上权值积的和，而不能求和的和
- 考虑按三进制逐位处理，若一条边边权为 i，则我们将其视为 x^i
- 考虑这种情况下跑矩阵树，则得出的是一个次数不超过 $2n$ 的多项式，x^i 的系数 A_i 即当前位边权和为 i 的方案数
- $A_i \times (i \bmod 3)$ 即它的贡献
- 但直接用多项式无法方便求解行列式
- 考虑到次数不大，我们取 $0 \sim 2n$ 共 $2n+1$ 个数代入多项式算出值，最后利用插值求解原多项式即可

生成树求和

- 容易想到矩阵树定理，但矩阵树定理只能求边上权值积的和，而不能求和的和
- 考虑按三进制逐位处理，若一条边边权为 i，则我们将其视为 x^i
- 考虑这种情况下跑矩阵树，则得出的是一个次数不超过 $2n$ 的多项式，x^i 的系数 A_i 即当前位边权和为 i 的方案数
- $A_i \times (i \bmod 3)$ 即它的贡献
- 但直接用多项式无法方便求解行列式
- 考虑到次数不大，我们取 $0 \sim 2n$ 共 $2n+1$ 个数代入多项式算出值，最后利用插值求解原多项式即可
- $O(n^4 \log c)$