树形 DP 选讲

顾逸宏

August 16, 2016

如何较好掌握 DP

如何较好掌握 DP

- DP 的**核心**在于它的状态

如何较好掌握 DP

- DP 的**核心**在于它的状态
- 状态 = { 状态的表示，状态的转移 }

如何较好掌握 DP

- DP 的**核心**在于它的状态
- 状态＝｛状态的表示，状态的转移｝
- DP 的**核心能力**是分类讨论＋归纳的能力

如何较好掌握 DP

- DP 的**核心**在于它的状态
- 状态 = { 状态的表示，状态的转移 }
- DP 的**核心能力**是分类讨论 + 归纳的能力
- 学习掌握基本类型：普通 DP，树形 DP，数位 DP，状压 DP，DP 的优化...

如何较好掌握 DP

- DP 的**核心**在于它的状态
- 状态 = { 状态的表示，状态的转移 }
- DP 的**核心能力**是分类讨论 + 归纳的能力
- 学习掌握基本类型：普通 DP，树形 DP，数位 DP，状压 DP，DP 的优化...
- 大量的练习：Topcoder ...

关于一些默认设定

关于一些默认设定

- x : 当前考虑的节点

关于一些默认设定

- x : 当前考虑的节点
- c : x 的儿子

关于一些默认设定

- x : 当前考虑的节点
- c : x 的儿子
- f : x 的父亲

关于一些默认设定

- x : 当前考虑的节点
- c : x 的儿子
- f : x 的父亲
- (x, y) : x 到 y 的路径

关于一些默认设定

- x : 当前考虑的节点
- c : x 的儿子
- f : x 的父亲
- (x, y) : x 到 y 的路径
- $w(x, y)$: x 到 y 的路径长度

概要

经典问题

经典问题

- 给定一个无向图 G，求图 G 的任意两个点之间最短路的最大值。

经典问题

- 给定一个无向图 G, 求图 G 的任意两个点之间最短路的最大值。
- 情形 1:G 是一个普通的图

经典问题

- 给定一个无向图 G，求图 G 的任意两个点之间最短路的最大值。
- 情形 1:G 是一个普通的图
- 情形 2:G 是树

经典问题

- 给定一个无向图 G，求图 G 的任意两个点之间最短路的最大值。
- 情形 1:G 是一个普通的图
- 情形 2:G 是树
- 情形 2-EX:G 是树，同时求出以每个点为起点的最长路

经典问题

- 给定一个无向图 G，求图 G 的任意两个点之间最短路的最大值。
- 情形 1:G 是一个普通的图
- 情形 2:G 是树
- 情形 2-EX:G 是树，同时求出以每个点为起点的最长路
- 情形 3:G 上只有一个环

经典问题

- 给定一个无向图 G，求图 G 的任意两个点之间最短路的最大值。
- 情形 1:G 是一个普通的图
- 情形 2:G 是树
- 情形 2-EX:G 是树，同时求出以每个点为起点的最长路
- 情形 3:G 上只有一个环
- 情形 4:G 是仙人掌

概要

暴力枚举

暴力枚举

- 枚举起点 i，然后做 SPFA，求最大值即可。

暴力枚举

- 枚举起点 i，然后做 SPFA，求最大值即可。
- Floyd，复杂度 $O(n^3)$

概要

贪心算法

贪心算法

- 随便取一个点 x，求出 G 中离 x 最远的点 y，再求出 G 中离 y 最远的点 z，y 到 z 的路径就是最长路径之一

贪心算法

- 随便取一个点 x，求出 G 中离 x 最远的点 y，再求出 G 中离 y 最远的点 z，y 到 z 的路径就是最长路径之一

证明

贪心算法

- 随便取一个点 x，求出 G 中离 x 最远的点 y，再求出 G 中离 y 最远的点 z，y 到 z 的路径就是最长路径之一

证明

反证法：假设存在一条更长的路径 (y', z')

贪心算法

- 随便取一个点 x，求出 G 中离 x 最远的点 y，再求出 G 中离 y 最远的点 z，y 到 z 的路径就是最长路径之一

证明

反证法：假设存在一条更长的路径 (y', z')
显然，y, z, y', z' 都是度为 1 的节点

贪心算法

- 随便取一个点 x，求出 G 中离 x 最远的点 y，再求出 G 中离 y 最远的点 z，y 到 z 的路径就是最长路径之一

证明

反证法：假设存在一条更长的路径 (y', z')
显然，y, z, y', z' 都是度为 1 的节点
分类讨论

贪心算法

- 随便取一个点 x，求出 G 中离 x 最远的点 y，再求出 G 中离 y 最远的点 z，y 到 z 的路径就是最长路径之一

证明

反证法：假设存在一条更长的路径 (y', z')
显然，y, z, y', z' 都是度为 1 的节点
分类讨论

- 情形 1:(y, z) 和 (y', z') 不相交

贪心算法

- 随便取一个点 x，求出 G 中离 x 最远的点 y，再求出 G 中离 y 最远的点 z，y 到 z 的路径就是最长路径之一

证明

反证法：假设存在一条更长的路径 (y', z')
显然，y, z, y', z' 都是度为 1 的节点
分类讨论

- 情形 1:(y, z) 和 (y', z') 不相交
- 情形 2:(y, z) 和 (y', z') 存在交点 p

动态规划

动态规划

- 取 1 为根，变成有根树

动态规划

- 取 1 为根，变成有根树
- 设 $L[x]$ 为以 x 为根的子树中的最长路

动态规划

- 取 1 为根，变成有根树
- 设 $L[x]$ 为以 x 为根的子树中的最长路
- 显然 $L[1]$ 即为答案

如何求 $L[x]$

分类讨论——两种路径

如何求 $L[x]$

分类讨论——两种路径

- 不经过 x 的路径：即 L[c] 的最大值

如何求 $L[x]$

分类讨论——两种路径

- 不经过 x 的路径：即 L[c] 的最大值
- 经过 x 的路径：设 $down[x]$ 为 x 往下走能走的最长路的长度，路径长度的最大值即为 $down[c]$ 中的最大值和次大值之和（这里次大值的初始值为 0）

如何求 $L[x]$

分类讨论——两种路径

- 不经过 x 的路径：即 L[c] 的最大值
- 经过 x 的路径：设 $down[x]$ 为 x 往下走能走的最长路的长度，路径长度的最大值即为 $down[c]$ 中的最大值和次大值之和（这里次大值的初始值为 0）

如何求 $down[x]$？

如何求 $L[x]$

分类讨论——两种路径

- 不经过 x 的路径：即 L[c] 的最大值
- 经过 x 的路径：设 $down[x]$ 为 x 往下走能走的最长路的长度，路径长度的最大值即为 $down[c]$ 中的最大值和次大值之和（这里次大值的初始值为 0）

如何求 $down[x]$？

- 即 $down[c] + w(x, c)$ 的最大值

概要

情形 2-EX

问题

G 是树，同时求出以每个点为起点的最长路

情形 2-EX

问题

G 是树，同时求出以每个点为起点的最长路

分类讨论

情形 2-EX

问题

G 是树，同时求出以每个点为起点的最长路

分类讨论

- 第一步往下走：即 $down[x]$

情形 2-EX

问题

G 是树，同时求出以每个点为起点的最长路

分类讨论

- 第一步往下走：即 $down[x]$
- 第一步往上走：设 $up[x]$ 为以 x 为起点，第一步往上走能走的最长路的长度

如何求 $up[x]$

分类讨论

第一步从 x 走到了 f,

如何求 $up[x]$

分类讨论

第一步从 x 走到了 f,

- 第二步往上走：即 $up[f]$

如何求 $up[x]$

分类讨论

第一步从 x 走到了 f,

- 第二步往上走：即 $up[f]$
- 第二步往下走（注意这里不能回到 x）: 如果 $down[f]$ 不是由 $down[x]$ 更新得来，那么就是 $down[f]$，如果是的话，就是 $down2[f]$

如何求 $up[x]$

分类讨论

第一步从 x 走到了 f,

- 第二步往上走：即 $up[f]$
- 第二步往下走（注意这里不能回到 x）：如果 $down[f]$ 不是由 $down[x]$ 更新得来，那么就是 $down[f]$，如果是的话，就是 $down2[f]$

$up[x]$ 为两者的最大值 $+w(x, f)$

如何求 $up[x]$

分类讨论

第一步从 x 走到了 f,

- 第二步往上走：即 $up[f]$
- 第二步往下走（注意这里不能回到 x）: 如果 $down[f]$ 不是由 $down[x]$ 更新得来，那么就是 $down[f]$，如果是的话，就是 $down2[f]$

$up[x]$ 为两者的最大值 $+w(x, f)$

如何求 $down2[x]$

$down[c] + w(x, c)$ 的次大值

如何求 $up[x]$

分类讨论

第一步从 x 走到了 f,

- 第二步往上走：即 $up[f]$
- 第二步往下走（注意这里不能回到 x）: 如果 $down[f]$ 不是由 $down[x]$ 更新得来，那么就是 $down[f]$，如果是的话，就是 $down2[f]$

$up[x]$ 为两者的最大值 $+w(x,f)$

如何求 $down2[x]$

$down[c]+w(x,c)$ 的次大值

$down1v[x]$

记录 $down[x]$ 是由哪个 c 转移过来的

概要

情形 3

问题

G 中只有一个环

情形 3

问题

G 中只有一个环

注意是**最短路的最大值**

情形 3

问题

G 中只有一个环

注意是**最短路的最大值**

bfs/dfs 一遍，找出环，找到所有的环上的节点 $cir[i]$

情形 3

问题

G 中只有一个环

注意是**最短路的最大值**

bfs/dfs 一遍，找出环，找到所有的环上的节点 $cir[i]$

分类讨论

情形 3

问题

G 中只有一个环

注意是**最短路的最大值**

bfs/dfs 一遍，找出环，找到所有的环上的节点 $cir[i]$

分类讨论

- 不经过环：和树上一样的做法，最后取 $L[cir[i]]$ 的最大值

情形 3

问题

G 中只有一个环

注意是**最短路的最大值**

bfs/dfs 一遍，找出环，找到所有的环上的节点 $cir[i]$

分类讨论

- 不经过环：和树上一样的做法，最后取 $L[cir[i]]$ 的最大值
- 经过环 :???

经过环 - 暴力

经过环 - 暴力

- 暴力枚举环上的两个点 x 和 y

经过环 - 暴力

- 暴力枚举环上的两个点 x 和 y
- 答案为 $down[x] + down[y] + min\{w1(x, y), w2(x, y)\}$

经过环 - 暴力

- 暴力枚举环上的两个点 x 和 y
- 答案为 $down[x] + down[y] + min\{w1(x, y), w2(x, y)\}$
- 复杂度 $O(k^2)$， k 为环长

经过环

经过环

- 破环成链，长度加倍，随便确定一个方向

经过环

- 破环成链，长度加倍，随便确定一个方向
- 枚举节点 x，考虑所有沿着方向走到 x 更短的 y，y 构成一个区间，假设此区间为 $[l_x, r_x]$

经过环

- 破环成链，长度加倍，随便确定一个方向
- 枚举节点 x，考虑所有沿着方向走到 x 更短的 y，y 构成一个区间，假设此区间为 $[l_x, r_x]$
- l_x 单调不降，r_x 单调不降

经过环

- 破环成链，长度加倍，随便确定一个方向
- 枚举节点 x，考虑所有沿着方向走到 x 更短的 y，y 构成一个区间，假设此区间为 $[l_x, r_x]$
- l_x 单调不降，r_x 单调不降
- 单调队列维护最值

经过环

- 破环成链，长度加倍，随便确定一个方向
- 枚举节点 x，考虑所有沿着方向走到 x 更短的 y，y 构成一个区间，假设此区间为 $[l_x, r_x]$
- l_x 单调不降，r_x 单调不降
- 单调队列维护最值
- 复杂度 $O(k)$，k 为环长

概要

情形 4

问题

G 是仙人掌，即任意两个环最多只有一个公共点。

情形 4

问题

G 是仙人掌，即任意两个环最多只有一个公共点。

进行一遍 dfs，求出 dfs 序

情形 4

问题

G 是仙人掌，即任意两个环最多只有一个公共点。

- 进行一遍 dfs，求出 dfs 序
- 对于每个节点 x 考虑条边 (x, y)

情形 4

问题

G 是仙人掌，即任意两个环最多只有一个公共点。

进行一遍 dfs，求出 dfs 序
对于每个节点 x 考虑条边 (x, y)

分类讨论

情形 4

问题

G 是仙人掌，即任意两个环最多只有一个公共点。

进行一遍 dfs，求出 dfs 序
对于每个节点 x 考虑条边 (x, y)

分类讨论

- 是树边：照常更新

情形 4

问题

G 是仙人掌，即任意两个环最多只有一个公共点。

进行一遍 dfs，求出 dfs 序
对于每个节点 x 考虑条边 (x, y)

分类讨论

- 是树边：照常更新
- 不是树边：???

环上的情况

环上的情况

- 按照 dfs 序从后往前做，一个环求答案+算 $down$ 值的顺序和这个环中 dfs 序最小的节点相关

环上的情况

- 按照 dfs 序从后往前做，一个环求答案+算 $down$ 值的顺序和这个环中 dfs 序最小的节点相关
- 环上的边不对 down 做更新

环上的情况

- 按照 dfs 序从后往前做，一个环求答案+算 *down* 值的顺序和这个环中 dfs 序最小的节点相关
- 环上的边不对 down 做更新
- 显然，一个环里面对后面的答案能够产生影响的，只有这个环中 dfs 序最小的节点。

核心代码

```
for (int k = c[x]; ~k; k = nxt[k]){
    int y = g[k];
    if (fa[y] == x && low[y] > idx[x]){
        ans = max(ans, down[y] + 1 + down[x]);
        down[x] = max(down[x], down[y] + 1);
    }
    if (fa[y] != x && idx[x] < idx[y]){
        N = 0;
        for (int i = y; i != fa[x]; i = fa[i])
                cir[++N] = i;
        updans();
        for (int i = 1; i < N; ++i)
            ckmax(down[x], down[cir[i]] + min(i,N-i));
    }
}
```

概要

1. 经典问题
   - 情形 1：普通的图
   - 情形 2：树形 DP
   - 情形 2-EX：down 与 up
   - 情形 3：环套树
   - 情形 4：仙人掌
2. 预备知识
3. 经典例题
4. 思考题

预备知识

预备知识

- 组合数，简单的去重

预备知识

- 组合数，简单的去重
- 启发式合并

预备知识

- 组合数，简单的去重
- 启发式合并
- 字符串 hash

预备知识

- 组合数，简单的去重
- 启发式合并
- 字符串 hash
- 概率论，离散型随机变量，期望

概要

1. 经典问题
   - 情形 1：普通的图
   - 情形 2：树形 DP
   - 情形 2-EX：down 与 up
   - 情形 3：环套树
   - 情形 4：仙人掌
2. 预备知识
3. 经典例题
4. 思考题

SPOJ MTREE

SPOJ MTREE

- 在一棵树上，求出所有路径边权积的和取模

SPOJ MTREE

- 在一棵树上，求出所有路径边权积的和取模
- $N \leq 10^5$

SPOJ GS

SPOJ GS

- 给定一棵树，每个点都有给定概率往其相邻点走，问从某一点到另一点期望走多少步。

SPOJ GS

- 给定一棵树，每个点都有给定概率往其相邻点走，问从某一点到另一点期望走多少步。
- $N \leq 10^5$

*CodeChef TAPAIR

*CodeChef TAPAIR

- 给定一个 N 个点的联通的无向图，问有多少种方案，删除 2 条边之后图不连通。

*CodeChef TAPAIR

- 给定一个 N 个点的联通的无向图，问有多少种方案，删除 2 条边之后图不连通。
- $N, M \leq 10^5$

SPOJ TREECST

SPOJ TREECST

- 将一棵树去掉一条边再加上一条边构成一棵树，使新的树直径最小，输出方案。

SPOJ TREECST

- 将一棵树去掉一条边再加上一条边构成一棵树，使新的树直径最小，输出方案。
- $N \leq 3 \times 10^5$

SPOJ TWOPATHS

SPOJ TWOPATHS

- 在一棵树上找两条严格不交的路径，使其长度积最大，输出这个积。

SPOJ TWOPATHS

- 在一棵树上找两条严格不交的路径，使其长度积最大，输出这个积。
- $N \leq 10^5$

CF202 Div1 E

CF202 Div1 E

- 给定一棵 n 个点的树，有 m 个点上有修道院。在今年，每个修道院里的修士们会列出离他们最远的修道院的名单，并且准备访问这么多个地方，邪恶的魔鬼想要炸掉一个没有修道院的点，如果能让一个点的人没有一个地方可以访问，他的愉悦值就会 $+1$。求最大的愉悦度和最大愉悦度的方案数。

CF202 Div1 E

- 给定一棵 n 个点的树，有 m 个点上有修道院。在今年，每个修道院里的修士们会列出离他们最远的修道院的名单，并且准备访问这么多个地方，邪恶的魔鬼想要炸掉一个没有修道院的点，如果能让一个点的人没有一个地方可以访问，他的愉悦值就会 $+1$。求最大的愉悦度和最大愉悦度的方案数。
- $m \le n \le 10^5$

NOI2012 Day2 迷失游乐园

NOI2012 Day2 迷失游乐园

- 给定一个 n 个点 m 条边的无向图，开始随机出现在某一点，每次等概率地往没走过的点走，求期望移动步数。

NOI2012 Day2 迷失游乐园

- 给定一个 n 个点 m 条边的无向图，开始随机出现在某一点，每次等概率地往没走过的点走，求期望移动步数。
- 注意 $m = n - 1$ 或 $m = n$，并且无向图连通

NOI2012 Day2 迷失游乐园

- 给定一个 n 个点 m 条边的无向图，开始随机出现在某一点，每次等概率地往没走过的点走，求期望移动步数。
- 注意 $m = n - 1$ 或 $m = n$，并且无向图连通
- 当 $m = n$ 时，图中唯一一个环大小 ≤ 30，$n \leq 10^5$

概要

1. 经典问题
   - 情形 1：普通的图
   - 情形 2：树形 DP
   - 情形 2-EX：down 与 up
   - 情形 3：环套树
   - 情形 4：仙人掌
2. 预备知识
3. 经典例题
4. 思考题

FoxConnection

FoxConnection

- 给出一棵 n 个节点的树，节点上可能有 1 只 fox，告诉你每个节点是否有 1 只 fox，让你移动这些 fox，使得每个节点最多只有一个 fox，有 fox 的节点相互连通，在这条件下使得移动的距离和最小。

FoxConnection

- 给出一棵 n 个节点的树，节点上可能有 1 只 fox，告诉你每个节点是否有 1 只 fox，让你移动这些 fox，使得每个节点最多只有一个 fox，有 fox 的节点相互连通，在这条件下使得移动的距离和最小。
- $n \leq 50$

InducedSubgraghs

InducedSubgraghs

- 给定一个 n 个节点的树，你现在要给 n 个节点重新标号，使得对于点集 $\{1, 2, \ldots k\}$,$\{2, 3, \ldots k+1\}$,...,$\{n-k+1, n-k+2, \ldots n\}$，他们在这棵树上的导出子树的点数都是 k。

InducedSubgraghs

- 给定一个 n 个节点的树，你现在要给 n 个节点重新标号，使得对于点集 $\{1, 2, \ldots k\}$,$\{2, 3, \ldots k + 1\}$,...,$\{n - k + 1, n - k + 2, \ldots n\}$，他们在这棵树上的导出子树的点数都是 k。
- $n \leq 50$

InducedSubgraghs

- 给定一个 n 个节点的树，你现在要给 n 个节点重新标号，使得对于点集 $\{1, 2, \ldots k\}$,$\{2, 3, \ldots k+1\}$,...,$\{n-k+1, n-k+2, \ldots n\}$，他们在这棵树上的导出子树的点数都是 k。
- $n \leq 50$
- $2 \times k \leq n$

InducedSubgraghs

- 给定一个 n 个节点的树，你现在要给 n 个节点重新标号，使得对于点集 $\{1, 2, \ldots k\}$,$\{2, 3, \ldots k+1\}$,...,$\{n-k+1, n-k+2, \ldots n\}$，他们在这棵树上的导出子树的点数都是 k。
- $n \leq 50$
- $2 \times k \leq n$
- $2 \times k \geq n$

solution for $2k \le n$

solution for $2k \le n$

- 那么，一定是中间一条长链，然后两边都挂着大小为 K 的子树，如下图。

solution for $2k \leq n$

- 那么，一定是中间一条长链，然后两边都挂着大小为 K 的子树，如下图。
- | [STA] - (k+1) - (k+2) - ... - (n-k) - [STB] |

solution for $2k \le n$

- 那么，一定是中间一条长链，然后两边都挂着大小为 K 的子树，如下图。
- | [STA] - (k+1) - (k+2) - ... - (n-k) - [STB] |
- 枚举链，找出 STA 和 STB。

solution for $2k \leq n$

- 那么，一定是中间一条长链，然后两边都挂着大小为 K 的子树，如下图。
- | [STA] - (k+1) - (k+2) - ... - (n-k) - [STB] |
- 枚举链，找出 STA 和 STB。
- STA 怎么染色，实际上就是所有的儿子都要比父亲小，问染色的方案数.

solution for $2k \le n$

- 那么，一定是中间一条长链，然后两边都挂着大小为 K 的子树，如下图。
- | [STA] - (k+1) - (k+2) - ... - (n-k) - [STB] |
- 枚举链，找出 STA 和 STB。
- STA 怎么染色，实际上就是所有的儿子都要比父亲小，问染色的方案数.
- 用 f[x][p] 表示 x 节点父亲为 p 的染色总方案数。

solution for $2k \leq n$

- 那么，一定是中间一条长链，然后两边都挂着大小为 K 的子树，如下图。
- | [STA] - (k+1) - (k+2) - ... - (n-k) - [STB] |
- 枚举链，找出 STA 和 STB。
- STA 怎么染色，实际上就是所有的儿子都要比父亲小，问染色的方案数.
- 用 f[x][p] 表示 x 节点父亲为 p 的染色总方案数。
- STB 类似于 STA。

solution for $2k \geq n$

solution for $2k \geq n$

- 中间有一个很大的联通块是 2k-n，联通块的一些节点连着一颗子树，要么是在 $[1, v]$ 里的，要么是在 $[n - v + 1, n]$ 里的.

solution for $2k \geq n$

- 中间有一个很大的联通块是 2k-n，联通块的一些节点连着一颗子树，要么是在 $[1, v]$ 里的，要么是在 $[n-v+1, n]$ 里的.
- 在这基础上做一个二维背包就可以了。

solution for $2k \geq n$

- 中间有一个很大的联通块是 2k-n，联通块的一些节点连着一颗子树，要么是在 $[1, v]$ 里的，要么是在 $[n-v+1, n]$ 里的.
- 在这基础上做一个二维背包就可以了。
- （细节）求的是：以 i 为联通块中的 root，所以要乘 $(2k-n)!$，但是这样也是错的，同一个联通块会被计算到 $(2k-n)$ 次，除去就可以了。

EagleInZoo

EagleInZoo

- 有一棵树，有 k 只老鹰要飞到这棵树上的某个节点休息，每只老鹰会按照以下的策略来确定自己休息的地方：

EagleInZoo

- 有一棵树，有 k 只老鹰要飞到这棵树上的某个节点休息，每只老鹰会按照以下的策略来确定自己休息的地方：
- 首先飞到根节点 0，然后，若到达的节点是空的，那么老鹰就停下。如果到达的节点不是空的，若他有儿子节点，那么随机选择一个，若没有，那么老鹰就只能飞走了

EagleInZoo

- 有一棵树，有 k 只老鹰要飞到这棵树上的某个节点休息，每只老鹰会按照以下的策略来确定自己休息的地方：
- 首先飞到根节点 0，然后，若到达的节点是空的，那么老鹰就停下。如果到达的节点不是空的，若他有儿子节点，那么随机选择一个，若没有，那么老鹰就只能飞走了
- 问第 k 只老鹰能够停在这棵树上的概率

EagleInZoo

- 有一棵树，有 k 只老鹰要飞到这棵树上的某个节点休息，每只老鹰会按照以下的策略来确定自己休息的地方：
- 首先飞到根节点 0，然后，若到达的节点是空的，那么老鹰就停下。如果到达的节点不是空的，若他有儿子节点，那么随机选择一个，若没有，那么老鹰就只能飞走了
- 问第 k 只老鹰能够停在这棵树上的概率
- $n \leq 50, k \leq 100$

EagleInZoo

EagleInZoo

- 分类讨论：$ans = \sum_{i=1}^{n} P(i)$，$P(i)$ 为第 k 轮的老鹰停在第 i 个节点的概率

EagleInZoo

- 分类讨论：$ans = \sum_{i=1}^{n} P(i)$，$P(i)$ 为第 k 轮的老鹰停在第 i 个节点的概率
- 我们设 $f[i][j]$ 表示到了第 i 轮，节点 j 是空的，但是 j 的父亲非空的概率。那么 $ans = \sum_{i=1}^{n} f[K-1][i] * reach[i]$，其中 $reach[i]$ 表示从节点 1 到节点 i 的概率

EagleInZoo

- 分类讨论：$ans = \sum_{i=1}^{n} P(i)$，$P(i)$ 为第 k 轮的老鹰停在第 i 个节点的概率
- 我们设 $f[i][j]$ 表示到了第 i 轮，节点 j 是空的，但是 j 的父亲非空的概率。那么 $ans = \sum_{i=1}^{n} f[K-1][i] * reach[i]$，其中 $reach[i]$ 表示从节点 1 到节点 i 的概率
- 转移很简单，总复杂度 O(NK)

Thanks

MAIL:sbullet@163.com