

D小水獭和随机树（简单版）

题目描述

小水獭正在观摩一棵 n 个节点的树，其中每个点的权值来自一个随机数生成器，树的结构也在代码中直接钦定。它想知道对于每个点来说，所有与其距离小于 k 的祖先中（包括其本身），权值最大点的权值是多少？它的 C++ 代码如下：

```
#include <bits/stdc++.h>
#define maxn 10000086

using namespace std;

unsigned seed;

unsigned Rand(){
    seed ^= seed << 13;
    seed ^= seed >> 7;
    seed ^= seed << 17;
    return seed;
}

int t, n, k;
int fa[maxn];
unsigned a[maxn];

int main(){
    scanf("%d", &t);
    while(t--){
        scanf("%d%d%u", &n, &k, &seed);
        for(int i = 2; i <= n; i++) fa[i] = i - 1;
        for(int i = 1; i <= n; i++) a[i] = Rand();
        unsigned ans = 0;
        for(int i = 1; i <= n; i++){
            unsigned mx = 0;
            int x = i;
            for(int j = 1; j <= k; j++){
                mx = max(mx, a[x]);
                if(x == 1) break;
                x = fa[x];
            }
            ans += mx ^ i;
        }
        printf("%u\n", ans);
    }
}
```

其中 `fa[i]` 表示编号为 `i` 的节点的父节点编号。

可惜这份代码 TLE 了，你能帮它修改一下代码，得到一份在本题数据范围限制下等价但速度更快的代码吗？

输入格式

第一行一个正整数 tt ($1 \leq t \leq 51 \leq t \leq 5$)，表示数据组数。

对于每组数据，一行三个整数 $n, k, seed$ ($1 \leq n \leq 1071 \leq n \leq 107$, $1 \leq k \leq n1 \leq k \leq n$, $1 \leq seed < 2321 \leq seed < 232$)，表示树的节点数、距离参数和初始随机种子。

输出格式

对于每组数据，按照小水獭原本代码的逻辑，输出一行一个非负整数。

题意理解

本题中的树，每个节点最多只有一个子节点，可以视为一条链。

分析错误代码，代码的目的是求出，对于每个点，所有与其距离小于 k 的祖先（包括本节点）的最大权值，再将权值与节点序号按位异或，求和得到答案。

错误原因

分析数据规模， $n = 10^7, k \leq n$ 。在原算法中，对于每一个节点，都遍历了距离此节点距离小于 k 的节点，因此时间复杂度为 $O(nk)$ ，超时了！

对时间复杂度优化——单调队列

为了优化时间复杂度，我们可以采用单调队列的方式，令时间复杂度降到 $O(n)$ ，即可通过本题。

单调队列的介绍

性质

1. 队列中的元素其对应原来的列表中的顺序必须是单调递增的。
2. 队列中元素大小必须是有序的

特性

与普通队列不同，单调队列不仅需要实现队首出队，还需要实现队尾出队。在 `c++` 中，可以利用 `deque` 数据结构实现

实现

利用单调队列完成本题，我们可以定义一个 `deque Q` 来存放节点序号。

1. 遍历每个节点，对于本节点，若队尾元素小于本节点，此元素必不是合法（权值不是本节点 k 距离内最大的），将队尾元素出队。
2. 之后需要获取，本节点 k 距离内的最大权值节点。需要判断队首元素，若不合格（距离大于等于 k ）则出队。
3. 最终读取第一个合法的节点将它与节点序号异或，累加入 `ans`。

样例解释

以样例来说

```
1
5 3 7
```

1 - 5 号节点的权值为：

```
3280921031, 3288427972, 3562315295, 3873329379, 1168094338
```

1号节点：

因为队列为空直接入队。判断队首 1号元素合法， $ans += (3288427972 \wedge 1)$

2号节点：

判断队尾，1号节点权值3280921031 小于3号节点 3288427972，因此1号节点出队。因为此时队列为空，2号节点入队。判断队首 2号元素合法， $ans += (3288427972 \wedge 2)$

3号节点：

判断队尾，2号节点权值 3288427972 小于3号节点 3562315295，因此2号节点出队。因为此时队列为空，3号节点入队。判断队首 3号元素合法， $ans += (3562315295 \wedge 3)$

4号节点：

判断队尾，3号节点权值 3562315295 小于4号节点 3873329379，因此3号节点出队。因为此时队列为空，4号节点入队。判断队首 4号元素合法， $ans += (3873329379 \wedge 4)$

5号节点：

判断队尾，4号节点权值 3873329379 大于 5号节点 1168094338，不出队。而4号节点有可能作为后续节点的最大权值，因此入队入队。判断队首 4 号元素合法， $ans += (3873329379 \wedge 5)$

AC代码

```
#include <bits/stdc++.h>
#define maxn 10000086

using namespace std;

unsigned seed;

unsigned Rand() {
    seed ^= seed << 13;
    seed ^= seed >> 7;
    seed ^= seed << 17;
    return seed;
}

int t, n, k;
int fa[maxn];
unsigned a[maxn];
```

```

deque<int> Q;
int main() {
    scanf("%d", &t);
    while (t--) {
        scanf("%d%d%u", &n, &k, &seed);
        for (int i = 2; i <= n; i++) fa[i] = i - 1;
        for (int i = 1; i <= n; i++) a[i] = Rand();
        unsigned ans = 0;
        Q.clear();
        for (int i = 1; i <= n; i++) {
            unsigned mx = 0;

            while(Q.size() && a[Q.back()] <= a[i] )
                Q.pop_back();

            Q.push_back(i);
            while (i - Q.front() >= k && Q.size() )
            {
                Q.pop_front();
            }
            mx = a[Q.front()];

            ans += mx ^ i;
        }
        printf("%u\n", ans);
    }
}

```

[更多资料——洛谷P1886 滑动窗口 / 【模板】单调队列](#)

[P1886 滑动窗口 / 【模板】单调队列 - 洛谷](#) | [计算机科学教育新生态 \(luogu.com.cn\)](#)