小水獭和随机树（困难版）

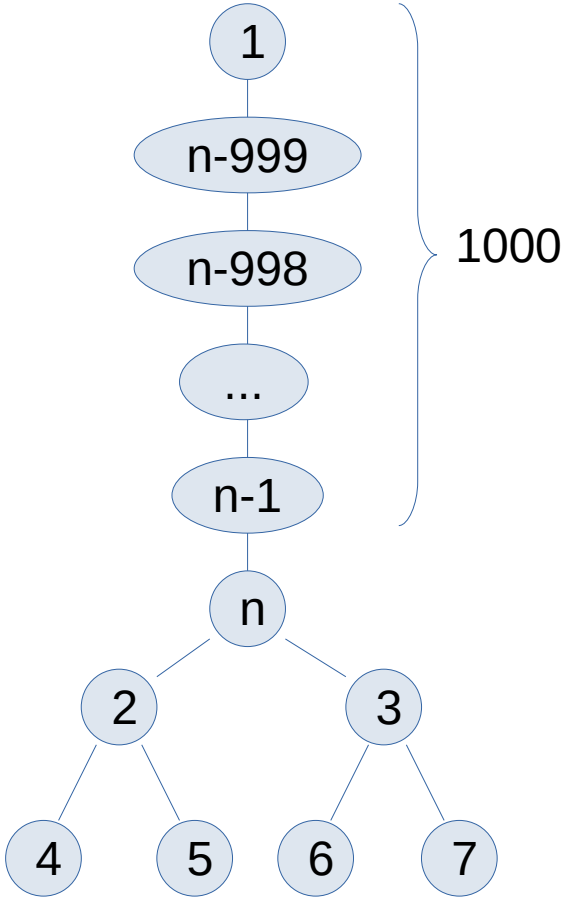
前置知识：单调队列，二叉树的前序遍历

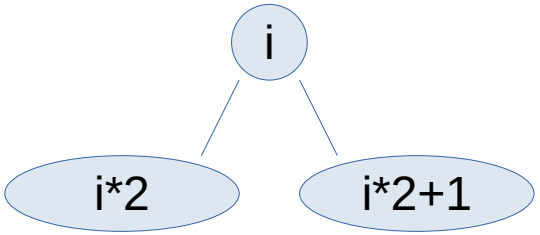
如果不知道什么是单调队列： https://oi-wiki.org/ds/monotonous-queue/

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：

$10^4 \leq n \leq 10^7,\ 1 \leq k \leq n$

```
scanf("%d%d%u", &n, &k, &seed);
for(int i = 4;i <= n - 1000;i++) fa[i] = i / 2;
for(int i = n - 998;i <= n;i++) fa[i] = i - 1;
fa[n - 999] = 1;
fa[2] = fa[3] = n;
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);
```

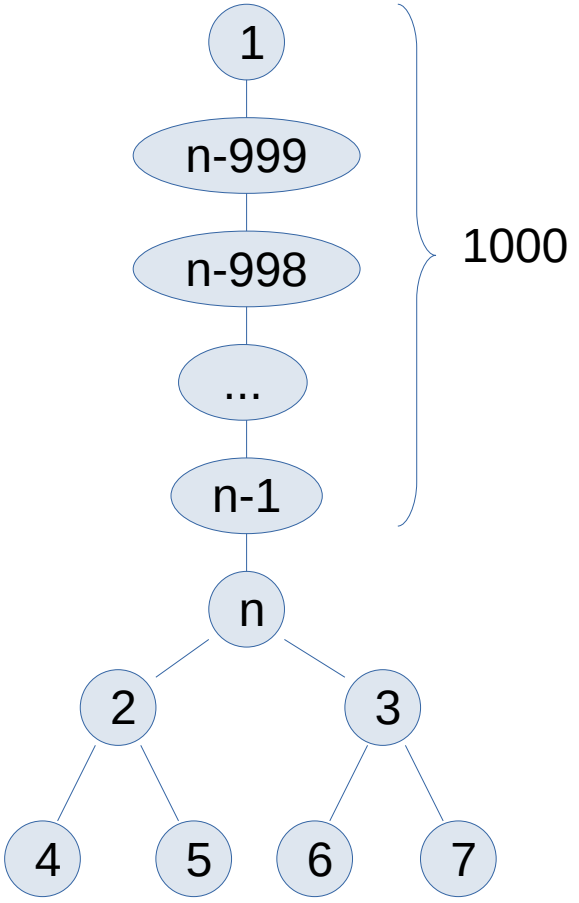




i != 1 && i <= n - 1000

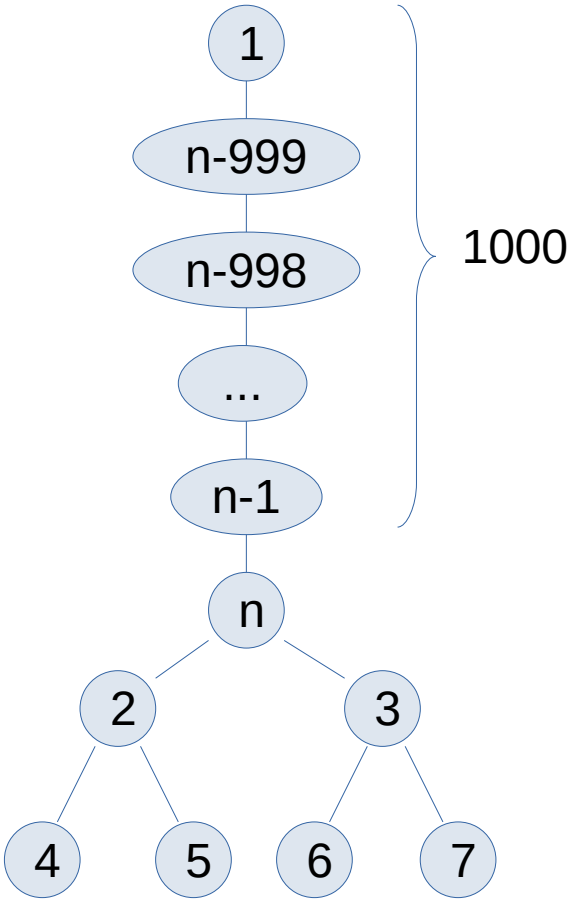
```
d[1] = 1001; // d[n] = 1001
for (int i = 2; i <= 10000000; i++) d[i] = d[i / 2] + 1;

int depth(int x) {
    if (x == 1) return 1;
    if (x > n - 1000) return x - n + 1001;
    return d[x];
}
```



```
l = 1; r = 1; q[1] = 1;
ans = a[1] ^ 1;
for (int i = n - 999; i <= n; i++) {
    while (l <= r && a[i] >= a[q[r]]) r--;
    q[++r] = i;
    if (depth(q[r]) - depth(q[l]) == k) l++;
    ans += a[q[l]] ^ i;
}
dfs(2);
dfs(3);
```

消除兄弟节点之间的影响
当一个节点退出 dfs 后，恢复单调队列状态至该节点加入队列之前的状态



用数组实现的单调队列，每当一个新元素加入队列时，发生变化的是头尾指针，且可能有一个元素被覆盖
如果我们保存新元素加入队列之前的头尾指针，保存新元素加入队列时替换掉的元素，在该节点退出 dfs 后即可 O(1) 恢复队列

```
void dfs(int i) {
    if (i > n - 1000) return;
    int lastl = l, lastr = r;
    while (l <= r && a[i] >= a[q[r]]) r--;
    int lasti = q[++r];
    q[r] = i;
    if (depth(q[r]) - depth(q[l]) == k) l++;
    ans += a[q[l]] ^ i;
    dfs(i * 2);
    dfs(i * 2 + 1);
    q[r] = lasti;
    l = lastl; r = lastr;
}
```

时间复杂度分析

- 每个节点加入队列的次数是 1
- 每个节点被覆盖的次数最大是以该节点为根的树的叶节点数
- 最坏时间复杂度 O(n(1000+logn))， ~~数据随机直接卡过去了~~