

A vibrant underwater scene featuring a large school of various fish swimming in clear, blue water. The fish are of different sizes and species, creating a dynamic and lively background. The lighting is bright, highlighting the silvery scales of the fish and the shimmering surface of the water.

Collecting Stamps 5 解説

解説担当者：blackyuki

Subtask 0

問題概要

問題概要

- N 頂点の木がある。
- 頂点の組 (s, g) は次の条件を満たす時良いペアと呼ぶ。
 - 木上のパスを $(s=i_0, i_1, i_2, \dots, i_d=g)$ とした時、
 $T_{\{i_j\}} \leq j$ を満たす $0 \leq j \leq d$ が存在し、かつ、 $d \leq D$
- $s=1, 2, \dots, N$ それぞれについて、 (s, g) が良いペアとなる g の個数を求めよ。

0

問題概要

制約

- $2 \leq N \leq 400000$
- $0 \leq D \leq N-1$
- $0 \leq T_i \leq N$

Subtask 1

$$D \leq 1$$

1

$D \leq 1$

- 頂点 s と頂点 g の距離を $\text{dist}(s,g)$ とする。
- $D \leq 1$ の時、 $\text{dist}(s,g) \leq D$ となる (s,g) の組は $O(N)$ 。
 - $\text{dist}(s,g) = 0$ は頂点数。
 - $\text{dist}(s,g) = 1$ は辺数の2倍。
- それぞれのペアについて愚直に判定すれば良い。
- 計算量は $O(N)$ 。

```
int N, D; cin >> N >> D;
vector<int> T(N), ans(N);
for(int i = 0; i < N; i++) {
    cin >> T[i];
    if(T[i] == 0) ans[i] ++;
}
for(int i = 0; i < N-1; i++) {
    int u, v; cin >> u >> v;
    u--; v--;
    if(D == 1) {
        if(T[u] <= 0 || T[v] <= 1) ans[u] ++;
        if(T[v] <= 0 || T[u] <= 1) ans[v] ++;
    }
}
for(int i = 0; i < N; i++) cout << ans[i] << '\n';
```

Subtask 2

パスグラフ、 $N \leq 3000$

2 パスグラフ、 $N \leq 3000$

- $s < g$ のケースだけ考える
 - $s = g$ は簡単、 $s > g$ は対称性により $s < g$ のケースに帰着できるので
- s から右方向に歩いて行った時、最初にスタンプを押せる街を m とすると、 (s, g) が良いペアとなる g は $g = m, m+1, \dots, s+D$ となる。
- 各 s について、 $O(N)$ かけて良い。
- 計算量は $O(N^2)$ 。

```
for(int s = 0; s < N; s++) {  
    int ans = 0;  
    for(int i = 0; i <= D && s + i < N; i++) if(T[s + i] <= i) {  
        ans += D - i + 1;  
        break;  
    }  
    for(int i = 0; i <= D && s - i >= 0; i++) if(T[s - i] <= i) {  
        ans += D - i + 1;  
        break;  
    }  
    if(T[s] == 0) ans--;  
    cout << ans << '\n';  
}
```

Subtask 3

$N \leq 3000$

3

$N \leq 3000$

- 小課題 2 と同様に、各 s ごとに $O(N)$ かけて良い。
 - DFS で木を探索する。
 - 以下の情報を持つ。
 - 現在の体力
 - スタンプを押したか (bool 値)
- 計算量は $O(N^2)$ 。

```
for(int s = 0; s < N; ++s) {
    int ans = 0;
    auto dfs = [&](auto &&self, int i, int p, int d, bool stamped) -> void {
        if(T[i] <= d) stamped = true;
        if(stamped) ans ++;
        if(d < D) {
            for(int to : G[i]) if(to != p) self(self, to, i, d+1, stamped);
        }
    }; dfs(dfs, s, -1, 0, false);
    cout << ans << '\n';
}
```

Subtask 4

パスグラフ

4 パスグラフ

- $s < g$ のケースのみ考える。
- 今後は逆に、各 g について、 s から g まで歩いた時に一度以上スタンプを押せるような最大の s を $dp[g]$ とする。
 - $dp[g] = \max(g - T[g], dp[g-1])$ で求められる。

4 パスグラフ

- g を固定した時、 (s, g) が良いペアとなる s は区間となる。
 - 具体的には $g-D, g-D+1, \dots, dp[g]$ となる。
 - imos 法で $ans[s]$ に区間加算すれば良い。
- $s > g$ についても同様にやる。
- 計算量は $O(N)$ 。

```

vector<int> sol(int N, int D, const vector<int>& T) {
    vector<int> ans(N), dp(N, -1);
    if(T[0] == 0) dp[0] = 0;
    for(int g = 1; g < N; g++) dp[g] = max(dp[g-1], g - T[g]);
    for(int g = 0; g < N; g++) if(g - D <= dp[g]){
        ans[max(0, g - D)] ++;
        if(dp[g] + 1 < N) ans[dp[g] + 1] --;
    }
    for(int i = 1; i < N; ++i) ans[i] += ans[i-1];
    return ans;
}

int main() {
    int N, D; cin >> N >> D;
    vector<int> T(N);
    for(int i = 0; i < N; i++) cin >> T[i];
    vector<int> ans = sol(N, D, T);
    reverse(T.begin(), T.end());
    vector<int> ans2 = sol(N, D, T);
    reverse(ans2.begin(), ans2.end());
    reverse(T.begin(), T.end());
    for(int i = 0; i < N; ++i) {
        ans[i] += ans2[i];
        if(T[i] == 0) ans[i]--;
        cout << ans[i] << '\n';
    }
}

```

Subtask 5,6

$D=N-1, N \leq 150000$

$N \leq 400000$

5

満点

- sub6 の解法を説明します。
 - 実装をサボったりすると sub5 だけ点が入るかもしれません。
- T_i が全部 0 の時を考える。
 - つまり、スタンプ台が最初から全て配置されている。
 - 各 s について、 s から距離 D 以下の頂点数を求めよ、という問題

5

満点

- 各 s について、 s から距離 D 以下の頂点数を求めよ、という問題
- 有名問題
- **重心分解**で $O(N \log N)$
 - 重心を通るパスについて $O(N)$ で処理できれば、再帰的に行うことで全体 $O(N \log N)$ 。

5

満点

- よって T_i が一般の場合も重心分解が必要になると予想される。
- 重心を通るパスだけ考えれば良い。
- それぞれの頂点について、以下の情報を計算しておく(全て dp で計算できる)。
 - 重心からの距離
 - その頂点を時刻 0 に出発し重心に向かう途中でスタンプを押せるか
 - 重心を時刻 t に出発しその頂点に向かう途中でスタンプを押せるような t の最小値

5

満点

- 全体計算量は $O(N\log N)$ 。
- 実装の細かい部分を詰めるのは難しいですが、頑張りましょう。
- 細かい部分をサボったり無駄な \log をつけたりすると sub5 だけ点が入るかもしれません。

得点分布

6

得点分布 (JOI)

• 72 点 : 2人

• 69 点 : 1人

• 31 点 : 15人

• 28 点 : 2人

• 21 点 : 1人

• 20 点 : 1人

• 17 点 : 2人

• 10 点 : 1人

• 3 点 : 1人

• 0 点 : 4人

6

得点分布 (JOIG)

• 48 点 : 2人

• 31 点 : 2人

• 26 点 : 1人

• 14点 : 1人

• 5 点 : 4人

• 0 点 : 5人