

Nộp bài CS112

Đề bài tóm tắt

Công ty có n nhân viên, nhân viên số 1 là CEO. Mỗi nhân viên i có khối lượng công việc w_i và (trừ CEO) có một quản lý p_i . Độ phức tạp công việc của nhân viên i được định nghĩa:

$$C_i = w_i + \text{số con trực tiếp của } i.$$

Khi sa thải nhân viên x , các con của x sẽ được chuyển thành con trực tiếp của p_x , và w_x được cộng vào w_{p_x} . Người số 1 không thể bị sa thải. Tìm số lượng người tối đa có thể sa thải sao cho mọi lúc mọi nhân viên còn lại đều có $C_i \leq m$.

Ý tưởng và thuật toán

Chiến lược đúng đắn là mô phỏng việc **sa thải các lá (leaf)** có thể bị sa thải:

- Lá có $\deg = 0$, nên $C_{\text{lá}} = w_{\text{lá}}$. Nếu $w_{\text{lá}} \leq m$ thì có thể sa thải lá đó.
- Sau khi sa thải, cha có thể trở thành lá mới; nếu thỏa điều kiện thì tiếp tục sa thải.

Thuật toán:

1. Xây dựng $\deg[i] = \text{số con trực tiếp ban đầu}$, $w[i]$ ban đầu.
2. Đưa vào hàng đợi tất cả các lá (trừ CEO) có $w[i] \leq m$.
3. Lặp: lấy từng nút u ra, nếu $w[u] + \deg[u] \leq m$ thì sa thải:
 - Cộng $w[u]$ cho cha, giảm $\deg[\text{parent}[u]]$.
 - Nếu cha trở thành lá mới và vẫn thỏa, thêm vào hàng đợi.

Độ phức tạp: $O(n)$.

Mã nguồn C++

```
1 #include <bits/stdc++.h>
2 using namespace std;
3
4 int main() {
5     ios::sync_with_stdio(false);
6     cin.tie(nullptr);
7 }
```

```

8      int n;
9      long long m;
10     cin >> n >> m;
11
12     vector<long long> w(n + 1);
13     for (int i = 1; i <= n; ++i) cin >> w[i];
14
15     vector<int> parent(n + 1, 0);
16     vector<int> deg(n + 1, 0);
17
18     for (int i = 2; i <= n; ++i) {
19         int p; cin >> p;
20         parent[i] = p;
21         deg[p]++;
22     }
23
24     vector<bool> removed(n + 1, false);
25     queue<int> q;
26
27     for (int i = 2; i <= n; ++i)
28         if (deg[i] == 0 && w[i] <= m) q.push(i);
29
30     int ans = 0;
31     while (!q.empty()) {
32         int u = q.front(); q.pop();
33         if (removed[u]) continue;
34         if (w[u] + deg[u] > m) continue;
35
36         removed[u] = true;
37         ans++;
38
39         int p = parent[u];
40         if (p == 0) continue;
41
42         w[p] += w[u];
43         deg[p]--;
44
45         if (p != 1 && deg[p] == 0 && w[p] + deg[p] <= m)
46             q.push(p);
47     }
48
49     cout << ans << "\n";
50     return 0;
51 }

```