

Bài toán Company - Chứng minh tính đúng đắn của thuật toán Greedy

1. Bài toán

Cho một cây với:

- Node 1 là CEO, không thể sa thải.
- Mỗi nhân viên i có khối lượng công việc w_i .
- Mỗi nhân viên i có độ phức tạp công việc:

$$C_i = w_i + \text{số lượng nhân viên do } i \text{ quản lý trực tiếp.}$$

Khi sa thải nhân viên i :

- Các con của i được chuyển sang quản lý bởi $p(i)$.
- Khối lượng công việc w_i được cộng vào $p(i)$.

Mục tiêu: **tối đa hóa số nhân viên có thể sa thải** sao cho mọi nhân viên còn lại đều thỏa mãn $C_i \leq m$.

2. Ý tưởng thuật toán

Thuật toán sử dụng DFS từ lá lên gốc:

1. Với mỗi node u , tính giá trị $\mathbf{dfs}(u)$ cho tất cả con v của u .
2. Lưu các giá trị này vào vector `child` và sắp xếp tăng dần.
3. Duyệt các giá trị trong `child`:
 - Nếu $ret + w - 1 \leq m$, sa thải con đó, cộng $w - 1$ vào ret và tăng biến đếm.
 - Nếu vượt m , dừng.
4. Trả về ret là độ phức tạp của subtree u sau khi tối đa hóa số con sa thải.

3. Chứng minh greedy cục bộ

Lemma 1 (Tối ưu cục bộ tại node u):

Giả sử node u có con v_1, v_2, \dots, v_k với giá trị $\text{dfs}(v_i)$ là khối lượng công việc sau khi tối đa hóa sa thải trong subtree v_i .

Bài toán: chọn các con để sa thải sao cho

$$ret = w_u + (\text{số con còn lại}) + \sum_{v \text{ sa thải}} (\text{dfs}(v) - 1) \leq m$$

và số con sa thải là tối đa.

Chứng minh:

- Đây là bài toán *chọn nhiều phần tử nhất mà tổng không vượt quá giới hạn*.
- Chiến lược **chọn các con có giá trị nhỏ nhất trước** luôn tối ưu:
 - Nếu chọn con lớn trước, khả năng sa thải thêm các con nhỏ sẽ giảm \rightarrow không tối ưu.
- Do đó, sắp xếp child tăng dần và chọn dần là tối ưu cục bộ.

4. Chứng minh tối ưu toàn cục

Lemma 2 (Bottom-up DFS đảm bảo tối ưu toàn cây):

- Mỗi node u chỉ phụ thuộc vào độ phức tạp của các subtree con đã xử lý.
- Vì mọi subtree con đã được tối ưu cục bộ (Lemma 1), quyết định tại node u là tối đa số con có thể sa thải mà không vượt m .
- DFS từ lá lên gốc đảm bảo:
 - Không vi phạm ràng buộc $C_i \leq m$.
 - Tổng số nhân viên bị sa thải toàn cây là tối đa.

5. Kết luận

Thuật toán DFS + Greedy (sắp xếp child tăng dần + chọn dần) là **đúng đắn**. Kết quả trả về là số nhân viên tối đa có thể sa thải mà vẫn đảm bảo $C_i \leq m$ cho mọi nhân viên còn lại:

Thuật toán trả về số nhân viên bị sa thải tối đa, thỏa ràng buộc độ phức tạp công việc.