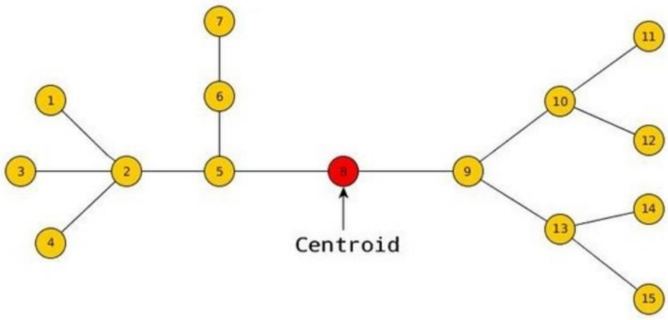
CENTROID DECOMPOSITION
(Phân rã trọng tâm trên cây)

I. Centroid:

Cho một cây N đỉnh, một centroid là đỉnh mà khi xóa nó đi sẽ tạo thành các cây mới, với điều kiện mỗi cây có không quá $\left[\frac{N}{2}\right]$ đỉnh.

Ví dụ:



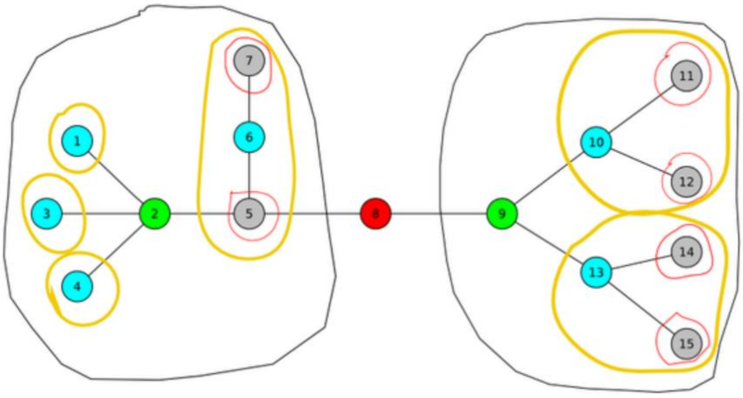
Định lí (Jordan, 1986): Cho một cây n đỉnh, tồn tại một đỉnh sao cho khi xóa đỉnh này cây sẽ được chia thành hai thành phần, mỗi thành phần có nhiều nhất N/2 đỉnh.

Chứng minh: Chọn một đỉnh *v* bất kì trên cây, nếu v thỏa mán tính chất của centroid thì chúng ta hoàn thành, ngược lại tồn tại một (và chỉ một) thành phần có nhiều hơn *N/2* đỉnh. Bây giờ, chúng ta xét các đỉnh *u* kề với *v* trong thành phần đó và áp dụng lập luận tương tự cho *u*. Chúng ta tiếp tục lập lại các thao tác trên cho đến khi tìm được đỉnh thỏa mãn. Ngoài ra, chúng ta không bao giờ quay lại bất kì đỉnh đã thăm nào, bởi vì các thành phần chứa chúng phải có ít hơn *N/2* đỉnh. Vì số đỉnh là hữu hạn và chúng ta thăm mỗi đỉnh nhiều nhất một lần nên các thao tác này phải kết thúc và vì thế centroid phải tồn tại.

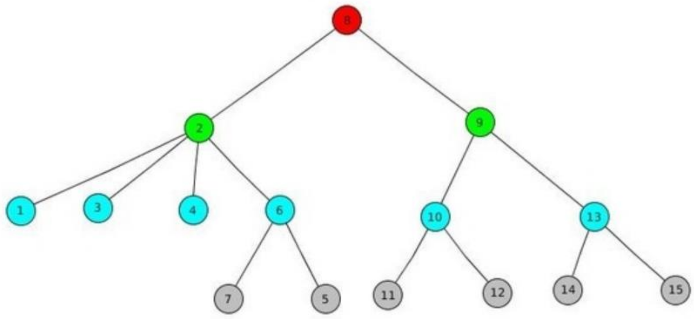
Tìm centroid của một cây: Một cách để tìm ra centroid là chọn một đỉnh góc *v* tùy ý, thực hiện thuật toán DFS để tính số nút con của cây con gốc *v*. Và chuyển sang gốc của cây con có kích thước lớn nhất,cho đến khi tìm được đỉnh mà nó không có cây con nào có kích thước lớn hơn *N/2*. Đỉnh đó là centroid cần tìm.

II. Centroid Tree

Khi tháo gỡ centroid, cây góc sẽ bị phân rã thành nhiều cây khác nhau. Mỗi cây có không nhiều hơn *N/2* đỉnh. Ta sẽ dùng centroid này để làm gốc cho Centroid Tree, đệ quy để phân rã các cây mới hình thành và dùng centroid của các cây này làm con của gốc Centroid Tree. Cuối cùng chúng ta có được một Centroid Tree từ cây đã cho ban đầu. Ví dụ:



Centroid Tree sẽ trông như hình dưới đây:



Các tính chất của Centroid Tree:

1. **Centroid Tree chứa tất cả N đỉnh của cây ban đầu.**
    - Vì mỗi nút sẽ trở thành centroid của các cây nhỏ (có thể cây chỉ là một đỉnh duy nhất). Vì thế centroid tree sẽ chứa tất cả *N* đỉnh của cây ban đầu.
2. **Centroid Tree có chiều cao lớn nhất là O(logN).**
    - Bởi vì tại mỗi bước, cây mới được tạo thành bằng các xóa centroid có n nhiều nhất *N/2* đỉnh, độ sâu cao nhất là *logN*, Nên độ cao lớn nhất mà centroid tree đạt được là *logN*.
3. **Chọn 2 đỉnh A và B bất kì trong đồ thị ban đầu thì đường đi giữa chúng có thể được chia thành A -> C và C -> B với C là LCA(Cha chung gần nhất của A và B trên Centroid Tree).**

III. Code xây dựng centroid tree.

```
void DFS_size (int u, int dad) {
    s[u] = 1;
    for (int v : adj[u])
        if (v != dad && !xoa[v]) {
            Pr[v] = u;
            DFS_size (v, u);
            s[u] += s[v];
        }
}

void Centroid_decompose (int r, int dad) {
    Pr[r] = 0;
```

```
    DFS_size (r, 0);
    int siz = s[r], smax;
    while (1) {
        smax = siz - s[r];
        int u = 0;
        for (int v : adj[r])
            if (!xoa[v]) {
                if (Pr[v] == r && smax < s[v]) {
                    smax = s[v];
                    u = v;
                }
            }
        if (smax <= siz / 2)
            break;
        r = u;
    }


    // Xu ly cac duong di qua r

    // Xoay cay
    xoa[r] = 1;
    for (int u : adj[r])
        if (!xoa[u]) {
            Centroid_decompose (u, r);
        }
}
```