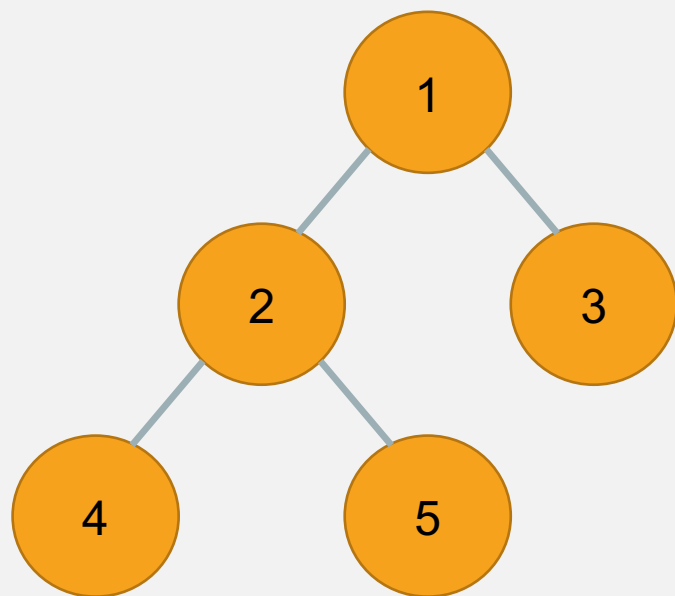


求子樹節點數量

林昱求 12.08.2020

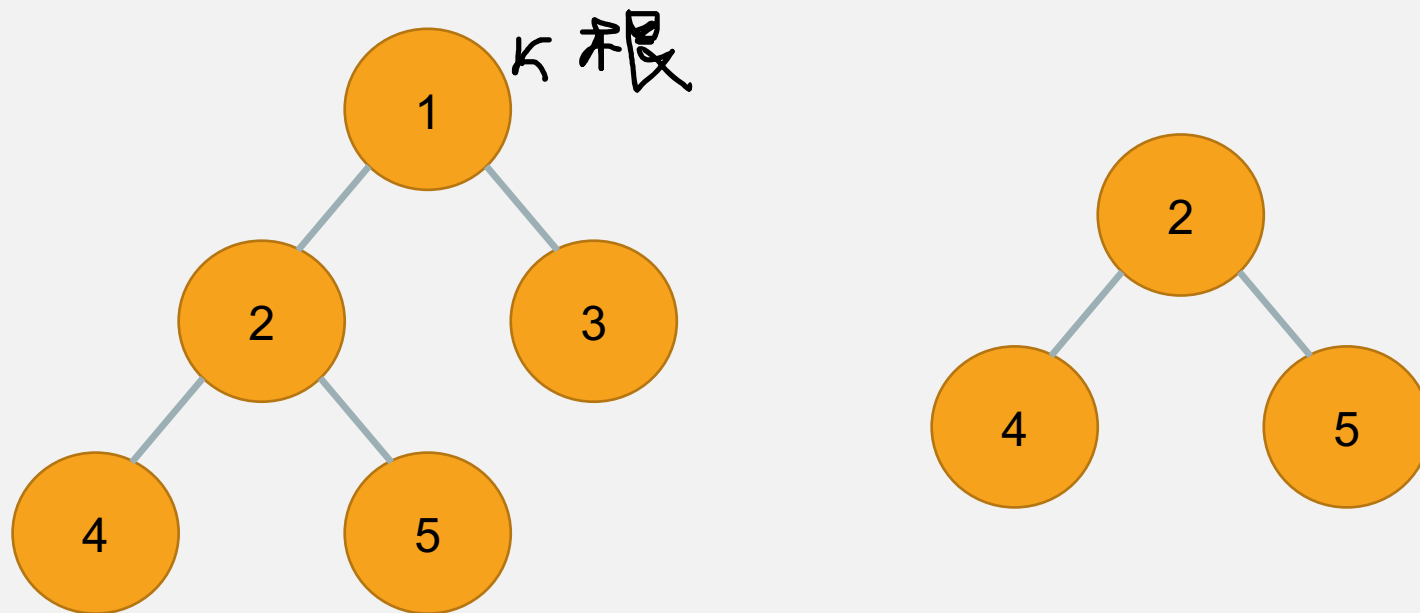
名詞定義



子代： u 節點的子代定義為，從根節點往下走經過 u 之後的所有節點。例如：2的子代有4, 5

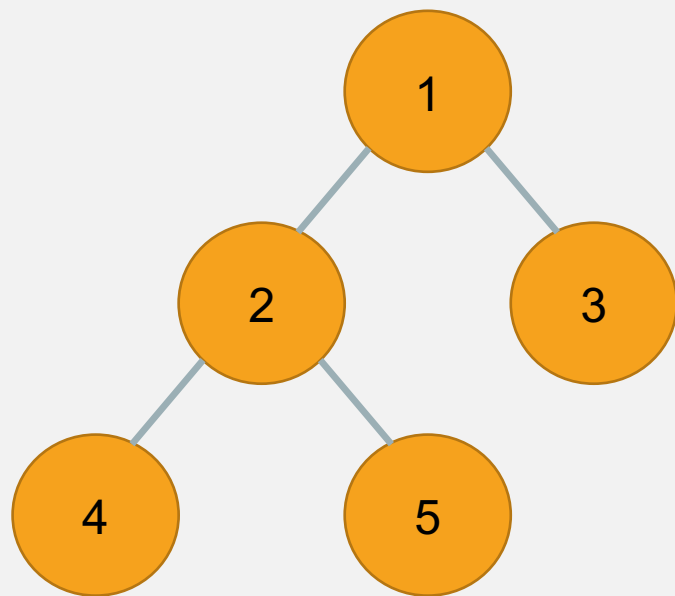
子樹：以 u 點為樹根的子樹定義為， u 自己和所有 u 子代的集合

名詞定義



右圖為左圖以2為根節點的子樹

題目說明



符號定義(我這邊自己定義的符號):

$\text{sub_cnt}(u)$ 為以 u 為根節點的子樹的節點數量。

例如：

$\text{sub_cnt}(2) = 3$ // 2, 4, 5 三個點

$\text{sub_cnt}(1) = 5$ // 1, 2, 3, 4, 5

DFS複習

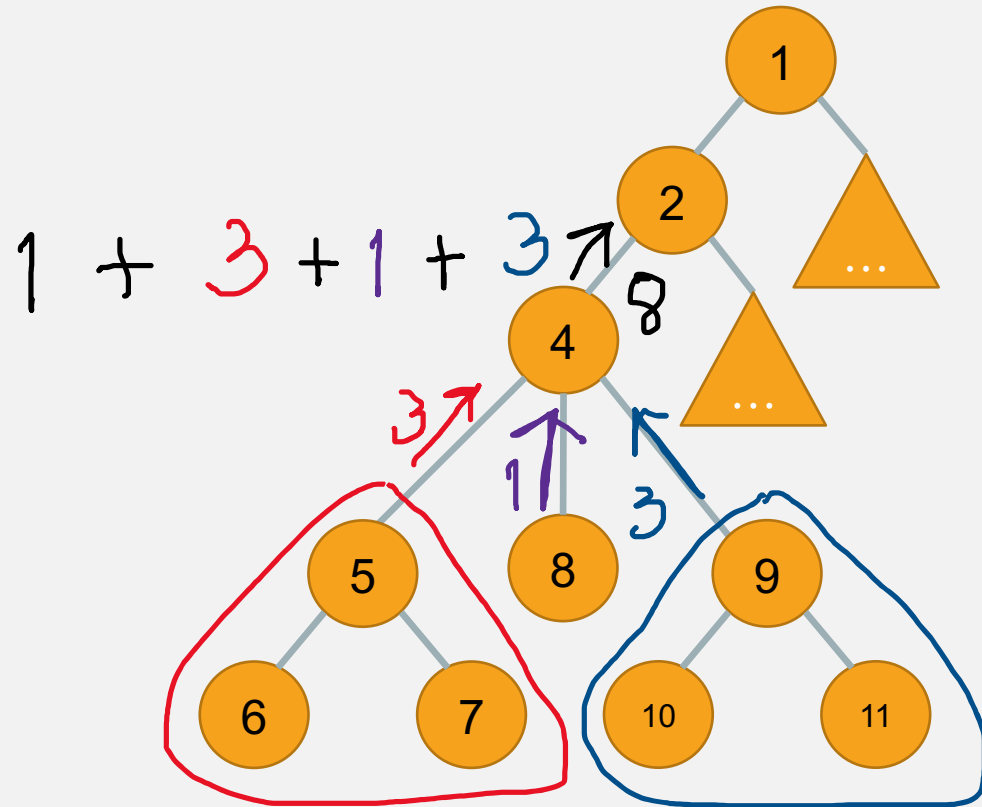
```
int visit[maxn] {}; // 是否走訪過
vector<int> e[maxn];
// e[u] = {v1, v2, v3 ...} u 到 v1, v2, v3...有邊
void dfs(int u) {
    // cout << u << ' ';
    for(int nxt : e[u]) if(visit[nxt] == false) {
        dfs(nxt);
    }
}
```

樹上DFS複習

樹沒有環，從根節點開始走，下一個點一定是子節點(**nxt**)
因此判斷下一個點不是父節點(**pre**)，就可以繼續走訪
樹上**dfs**可以省略掉**visit**這個陣列
p.s.以上變數都是我自己習慣用的而已

```
vector<int> e[maxn];  
// e[u] = {v1, v2, v3 ...} u 到 v1, v2, v3...有邊  
  
void dfs(int u, int pre) {  
    // cout << u << ' ' ;  
    for(int nxt : e[u]) if(nxt != pre) {  
        dfs(nxt);  
    }  
}
```

計算子樹數量節點數量



計算完當前子樹的節點數列後，回傳給父節點
 $\text{sub_cnt}(4) =$
 $\text{sub_cnt}(5) + \text{sub_cnt}(8) + \text{sub_cnt}(9) + 1$

最終程式碼

```
dfs(root, root); // dfs一次就能建好sub_cnt[]的表
cout << sub_cnt[query_node] << endl; // 印出詢問節點的sub_cnt()
```

```
vector<int> e[maxn];
int sub_cnt[maxn] {};
void dfs(int u, int pre) {
    int sub_cnt[u] = 1; // 自己算一個
    for(int nxt : e[u]) if(nxt != pre) {
        dfs(nxt);
        sub_cnt[u] += sub_cnt[nxt];
    }
}
```