

DFS-深度优先搜索

可用于遍历树或者图的搜索算法

也可用于计算树节点的权重和

C++

```
1  vector<int> a; // 记录每次排列
2  vector<int> book; //标记是否被访问
3
4  void DFS(int cur, int k, vector<int>& nums){
5      if(cur == k){ //k个数已经选完, 可以进行输出等相关操作
6          for(int i = 0; i < cur; i++){
7              printf("%d ", a[i]);
8          }
9          return ;
10     }
11     for(int i = 0; i < k; i++){ //遍历 n个数, 并从中选择k个数
12         if(book[nums[i]] == 0){ //若没有被访问
13             a.push_back(nums[i]); //选定本输, 并加入数组
14             book[nums[i]] = 1; //标记已被访问
15             DFS(cur + 1, n, nums); //递归, cur+1
16             book[nums[i]] = 0; //释放, 标记为没被访问, 方便下次引用
17             a.pop_back(); //弹出刚刚标记为未访问的数
18         }
19     }
20 }
21
```

dfs 递归计算权重和:

Python

```
1  # child 是一个存有 节点 左右节点id 的list
2  # [[1, 2], [0, 0], [3, 4], [0, 0], [0, 0]]
3
4  def dfs(u):
5      wsum = weight.copy()
6      visited[u] += 1
7      for i in child[u]:
8          if visited[u] != -1 and i != 0:
9              wsum[u] += dfs(i)
10     return wsum[u]
```