DFS 模版

```
#递归写法
visited = set()

def dfs(node, visited):
if node in visited: # terminator
        # already visited
        return

        visited.add(node)

        # process current node here.
        ...
        for next_node in node.children():
                if not next_node in visited:
                        dfs(next_node, visited)
```

非递归写法

```
def DFS(self, tree):

        if tree.root is None:
                return []

        visited, stack = [], [tree.root]

        while stack:
                node = stack.pop()
                visited.add(node)

                process (node)
                nodes = generate_related_nodes(node)
                stack.push(nodes)

        # other processing work
        ...
```