

暨南大学本科实验报告专用纸(附页)

树上 dfs (基础信息)

课程名称 数据结构 成绩评定
实验项目名称 树上 dfs (基础信息) 指导老师 干晓聪
实验项目编号 06 实验项目类型 设计性 实验地点 数学系机房
学生姓名 郭彦培 学号 2022101149
学院 信息科学技术学院 系 数学系 专业 信息管理与信息系统
实验时间 2024 年 6 月 13 日上午 ~ 2024 年 7 月 13 日中午

1. 实验目的

在树上进行 dfs

2. 实验环境

计算机: PC X64

操作系统: Windows + Ubuntu20.0LTS

编程语言: C++: GCC std20

IDE: Visual Studio Code

3. 程序原理

统计树每个节点的子树大小、节点深度 与 子树节点权值和。

对于节点 i , 其子节点列表 S_i , 父节点 p_i , 有状态转移:

$$\text{subTreeSize}_n = \sum_{\{ \text{all } i \in S_n \}} \text{subTreeSize}_i \quad (1)$$

$$\text{depth}_n = \text{depth}_{p_n} + 1 \quad (2)$$

$$\text{subTreeSum}_n = \sum_{\{ \text{all } i \in S_n \}} \text{subTreeSum}_i + \text{Wei}_n \quad (3)$$

4. 程序代码

4.1. dfs.cpp

```
1  #include <iostream>
2  #include <vector>
3  #include <stdlib.h>
4  #include <map>
5  #include <algorithm>
6
7  using namespace std;
8  using pii = pair<int,int>;
9
10 #define int long long
11 #define pb push_back
12 #define F first
13 #define S second
14 #define all(x) x.begin(),x.end()
15 #define loop(i,n) for(int i = 0; i < n; i++)
16
17 const int mod = 1e9 + 7;
18 const int INF = 1e18;
19
20 // 子树大小 节点深度 子树节点值和
21 void solve()
22 {
23     int n;
24     cin >> n;
25     vector<vector<int>> cnj(n+1);
26     vector<int> wei(n+1);
27     vector<int> subTreeSize(n+1),depth(n+1),subTreeSum(n+1);
28     loop(i,n-1)
29     {
30         int u,v;
31         cin >> u >> v;
32         cnj[u].pb(v);
33         cnj[v].pb(u);
34     }
35     loop(i,n) cin >> wei[i+1];
36     auto dfs = [&](auto self,int p,int f) -> pii {
37         int size = 1;
38         int sum = wei[p];
39         depth[p] = depth[f] + 1;
40         for(auto &x : cnj[p])
41         {
42             if(x == f) continue;
43             auto [subSize,subSum] = self(self,x,p);
44             size += subSize;
45             sum += subSum;
46         }
47         subTreeSize[p] = size;
```

```
48         subTreeSum[p] = sum;
49         return {size,sum};
50     };
51     dfs(dfs,1,0);
52     cout << "subTreeSize \n";
53     loop(i,n) cout << subTreeSize[i+1] << " ";
54     cout << "\n";
55     cout << "depth \n";
56     loop(i,n) cout << depth[i+1] << " ";
57     cout << "\n";
58     cout << "subTreeSum \n";
59     loop(i,n) cout << subTreeSum[i+1] << " ";
60     cout << "\n";
61 }
62
63 signed main()
64 {
65     // std::ios::sync_with_stdio(false);
66     // std::cin.tie(nullptr);
67     // std::cout.tie(nullptr);
68
69     int T = 1;
70     cin >> T;
71     while(T--)
72         solve();
73     return 0;
74 }
```

5. 测试数据与运行结果

运行上述 `_PRIV_TEST.cpp` 测试代码中的正确性测试模块，得到以下内容：

```
2
8
1 2
1 3
1 4
3 5
3 6
6 7
6 8
9 4 3 1 15 7 3 7
subTreeSize
8 1 5 1 1 3 1 1
depth
1 2 2 2 3 3 4 4
```

暨南大学本科实验报告专用纸(附页)

```
subTreeSum
49 4 35 1 15 17 3 7
6
1 2
2 3
3 4
4 5
5 6
1 1 4 5 1 4
subTreeSize
6 5 4 3 2 1
depth
1 2 3 4 5 6
subTreeSum
16 15 14 10 5 4
```

可以看出，代码运行结果与预期相符，可以认为代码正确性无误。