

# PAT甲级 2023年spring考试 Tree of Love

[https://blog.csdn.net/m0\\_50617544/article/details/129472724](https://blog.csdn.net/m0_50617544/article/details/129472724)

九是否随机的称呼 于 2023-05-03 18:22:56 发布

考试的时候最后一题花了近两个小时，然后最后一个测试点3分没有做出来，后来又花了一个上午的时间，还是没找到reason。最后下午的时候发现是输出outer contour的问题，就AC了。判断是否对称要层序遍历，存在节点就vec保存1，不存在vec保存0，若是该层的vec在reverse前后都相等，就是对称的。

判断是否是tree of love，要depth\_first保存叶子节点，然后进行判断。

输出的Outer contour，保存叶子节点要放在最前面，然后是最左侧的节点，最后是后序遍历的最右侧节点。最左侧和最右侧分别使用先left后right、先right后left的顺序进行递归遍历，以及不要重复，使用unordered\_map来记录是否访问过。

update AC codes

```
1  #include<iostream>
2  #include<vector>
3  #include<unordered_map>
4  #include<algorithm>
5  #include<queue>
6  using namespace std;
7  int in[100006], pot[100006], pre[100006], nextlevel, length;
8  struct nod {
9      nod *l = NULL, *r = NULL;
10     int val;
11 };
12 nod *recursion(nod *rot, int postroot, int inl, int inr) {
13     if(inl > inr) return NULL;
14     if(rot==NULL) {
15         rot = new(nod);
16         rot->val = pot[postroot];
17     }
18     int ind = inl;
19     while(ind <= inr && in[ind]!=pot[postroot]) ind++;
20     rot->l = recursion(rot->l, postroot - (inr - ind+1), inl, ind - 1);
21     rot->r = recursion(rot->r, postroot - 1, ind + 1, inr);
```

内容来源: csdn.net

作者昵称: 九是否随机的称呼

原文链接: [https://blog.csdn.net/m0\\_50617544/article/details/129472724](https://blog.csdn.net/m0_50617544/article/details/129472724)

作者主页: [https://blog.csdn.net/m0\\_50617544](https://blog.csdn.net/m0_50617544)

```

22     return rot; 23 }
23
24 struct nod2 {
25     nod *nd;
26     int level;
27 };
28 vector<int> v[100006], v0[100006];
29 unordered_map<nod *, int> rev;
30 bool width_first_symmetric(nod *rot) {
31     queue<nod2> q;
32     q.push({rot, 0});
33     nod2 nd2;
34     v[0].push_back(1);
35     v0[0].push_back(1);
36     rev[rot] = 0;
37     while(!q.empty()) {
38         nd2 = q.front();
39         rev[nd2.nd] = nd2.level;
40         nextlevel = nd2.level+1;
41         q.pop();
42         if(nd2.nd->l!=NULL) {
43             v[nextlevel].push_back(1);
44             v0[nextlevel].insert(v0[nextlevel].begin(), 1);
45             q.push(nod2{nd2.nd->l, nextlevel});
46         }else{
47             v[nextlevel].push_back(0);
48             v0[nextlevel].insert(v0[nextlevel].begin(), 0);
49         }
50         if(nd2.nd->r!=NULL) {
51             v[nextlevel].push_back(1);
52             v0[nextlevel].insert(v0[nextlevel].begin(), 1);
53             q.push(nod2{nd2.nd->r, nextlevel});
54         }else{
55             v[nextlevel].push_back(0);
56             v0[nextlevel].insert(v0[nextlevel].begin(), 0);
57         }
58     }
59     int kk = 9;
60     for(int i = 0; i < 100006; i++) {

```

内容来源: csdn.net

作者昵称: 九是否随机的称呼

原文链接: [https://blog.csdn.net/m0\\_50617544/article/details/129472724](https://blog.csdn.net/m0_50617544/article/details/129472724)

作者主页: [https://blog.csdn.net/m0\\_50617544](https://blog.csdn.net/m0_50617544)

```

61         if(v0[i]!=v[i]) { 62 |         kk = -9;
62             break;
63         }
64     }
65 }
66 if(kk < 0) return false;
67 else return true;
68 }
69 vector<int> vec, lev;
70 unordered_map<nod *, bool> tat;
71 bool leftmost = true, rightmost = true;
72 void depth_left_order(nod *rot) {
73     if(rot==NULL) {
74         leftmost = false;
75         return;
76     }
77     int level = rev[rot];
78     if(rot->l==NULL && rot->r==NULL && tat[rot]==false) {
79         vec.push_back(rot->val);
80         lev.push_back(level);
81         tat[rot] = true;
82     }
83     if(leftmost==true && tat[rot]==false) {
84         vec.push_back(rot->val);
85         tat[rot] = true;
86     }
87     depth_left_order(rot->l);
88     depth_left_order(rot->r);
89 }
90 void depth_right_order(nod *rot) {
91     if(rot==NULL || rightmost==false) {
92         rightmost = false;
93         return;
94     }
95     if(rightmost==true && tat[rot]==false) {
96         vec.insert(vec.begin() + length, rot->val);
97         tat[rot] = true;
98     }
99     depth_right_order(rot->r);
100    depth_right_order(rot->l);

```

内容来源: csdn.net

作者昵称: 九是否随机的称呼

原文链接: [https://blog.csdn.net/m0\\_50617544/article/details/129472724](https://blog.csdn.net/m0_50617544/article/details/129472724)

作者主页: [https://blog.csdn.net/m0\\_50617544](https://blog.csdn.net/m0_50617544)

```

101 }
102 | bool judge_tree love(){
103     int ll = 0, y, z, w, len = lev.size(), kl = 9;
104     if(ll+1 < len && lev[ll+1] < lev[ll]) kl = -9;
105     while(ll < len && lev[ll] >= lev[ll-1]) ll++;
106     y = ll;
107     if(y < len && lev[y] > lev[y-1]) kl = -9;
108     while(y < len && lev[y] <= lev[y-1]) y++;
109     z = y;
110     if(z < len && lev[z] < lev[z-1]) kl = -9;
111     while(z < len && lev[z] >= lev[z-1]) z++;
112     w = z;
113     if(w < len && lev[w] > lev[w-1]) kl = -9;
114     while(w < len && lev[w] <= lev[w-1]) w++;
115     if(w <= len && kl > 0) return true;
116     else return false;
117 }
118
119 int main(void) {
120     int i, N;
121     cin>>N;
122     nod nd, *rot=NULL;
123     for(i = 0; i < N; i++) scanf("%d", &in[i]);
124     for(i = 0; i < N; i++) scanf("%d", &pot[i]);
125     rot = recursion(rot, N-1, 0, N-1);
126     bool symmetric = width_first_symmetric(rot);
127     depth_left_order(rot);
128     length = vec.size();
129     depth_right_order(rot);
130     bool tree love = judge_tree love();
131     if(tree love==true && symmetric==true) printf("Yes\n");
132     else printf("No\n");
133     for(i = 0; i < vec.size(); i++) printf("%d%s", vec[i], i==(vec.size()-1)?" ":" ");
134     return 0;
135 }

```

内容来源: csdn.net

作者昵称: 九是否随机的称呼

原文链接: [https://blog.csdn.net/m0\\_50617544/article/details/129472724](https://blog.csdn.net/m0_50617544/article/details/129472724)

作者主页: [https://blog.csdn.net/m0\\_50617544](https://blog.csdn.net/m0_50617544)

```

1  #include<iostream>
2  #include<vector>
3  #include<unordered_map>
4  #include<algorithm>
5  #include<queue>
6  using namespace std;
7  int in[100006], pot[100006], pre[100006], nextlevel;
8  struct nod{
9      nod *l = NULL, *r = NULL;
10     int val;
11 };
12 nod *recursion(nod *rot, int postroot, int inl, int inr) {
13     if(inl > inr) return NULL;
14     if(rot==NULL) {
15         rot = new(nod);
16         rot->val = pot[postroot];
17     }
18     int ind = inl;
19     while(ind <= inr && in[ind]!=pot[postroot]) ind++;
20     rot->l = recursion(rot->l, postroot - (inr - ind+1), inl, ind - 1);
21     rot->r = recursion(rot->r, postroot - 1, ind + 1, inr);
22     return rot;
23 }
24 struct nod2{
25     nod *nd;
26     int level;
27 };
28 vector<int> v[100006], v0[100006];
29 vector<nod *> vk[100006];
30 unordered_map<int, int> ump;
31 unordered_map<nod *, int> rev;
32 bool width_first_symmetric(nod *rot){
33     queue<nod2> q;
34     q.push({rot, 0});
35     nod2 nd2;
36     v[0].push_back(1);
37     v0[0].push_back(1);
38     vk[0].push_back(rot);
39     ump[rot->val] = 0;

```

内容来源: csdn.net

作者昵称: 九是否随机的称呼

原文链接: [https://blog.csdn.net/m0\\_50617544/article/details/129472724](https://blog.csdn.net/m0_50617544/article/details/129472724)

作者主页: [https://blog.csdn.net/m0\\_50617544](https://blog.csdn.net/m0_50617544)

```

40 rev[rot] = 0;
41 while(!q.empty()) {
42     nd2 = q.front();
43     ump[nd2.nd->val] = nd2.level;
44     rev[nd2.nd] = nd2.level;
45     nextlevel = nd2.level+1;
46     q.pop();
47     if(nd2.nd->l!=NULL) {
48         v[nextlevel].push_back(1);
49         v0[nextlevel].insert(v0[nextlevel].begin(), 1);
50         vk[nextlevel].push_back(nd2.nd->l);
51         q.push(nod2{nd2.nd->l, nextlevel});
52     }else{
53         v[nextlevel].push_back(0);
54         v0[nextlevel].insert(v0[nextlevel].begin(), 0);
55         vk[nextlevel].push_back(NULL);
56     }
57     if(nd2.nd->r!=NULL) {
58         v[nextlevel].push_back(1);
59         v0[nextlevel].insert(v0[nextlevel].begin(), 1);
60         vk[nextlevel].push_back(nd2.nd->r);
61         q.push(nod2{nd2.nd->r, nextlevel});
62     }else{
63         v[nextlevel].push_back(0);
64         v0[nextlevel].insert(v0[nextlevel].begin(), 0);
65         vk[nextlevel].push_back(NULL);
66     }
67 }
68 int kk = 9;
69 for(int i = 0; i < 100006; i++) {
70     if(v0[i]!=v[i]) {
71         kk = -9;
72         break;
73     }
74 }
75 if(kk < 0) return false;
76 else return true;
77 }
78 vector<int> vec, leaves, lev;

```

内容来源: csdn.net

作者昵称: 九是否随机的称呼

原文链接: [https://blog.csdn.net/m0\\_50617544/article/details/129472724](https://blog.csdn.net/m0_50617544/article/details/129472724)

作者主页: [https://blog.csdn.net/m0\\_50617544](https://blog.csdn.net/m0_50617544)

```

79 unordered_map<nod *, bool> tat; 80 void depth_first(nod* rot) {
81     int level = rev[rot];
82     if(rot->l!=NULL && rot->r==NULL) {
83         if(tat[rot]==false) {
84             vec.push_back(rot->val);
85             lev.push_back(level);
86         }
87         tat[rot] = true;
88         return;
89     }
90     if(rot==vk[level][0]) {
91         if(tat[rot]==false) vec.push_back(rot->val);
92         tat[rot] = true;
93     }
94     if(rot->l!=NULL) depth_first(rot->l);
95     if(rot->r!=NULL) depth_first(rot->r);
96     if(rot==vk[level][vk[level].size() - 1]) {
97         if(tat[rot]==false) vec.push_back(rot->val);
98         tat[rot] = true;
99     }
100 }
101 bool judge_treelove(){
102     int ll = 0, y, z, w, len = lev.size(), kl = 9;
103     if(ll+1 < len && lev[ll+1] < lev[ll]) kl = -9;
104     while(ll < len && lev[ll] >= lev[ll-1]) ll++;
105     y = ll;
106     if(y < len && lev[y] > lev[y-1]) kl = -9;
107     while(y < len && lev[y] <= lev[y-1]) y++;
108     z = y;
109     if(z < len && lev[z] < lev[z-1]) kl = -9;
110     while(z < len && lev[z] >= lev[z-1]) z++;
111     w = z;
112     if(w < len && lev[w] > lev[w-1]) kl = -9;
113     while(w < len && lev[w] <= lev[w-1]) w++;
114     if(w <= len && kl > 0) return true;
115     else return false;
116 }
117 int main(void) {

```

内容来源: csdn.net

作者昵称: 九是否随机的称呼

原文链接: [https://blog.csdn.net/m0\\_50617544/article/details/129472724](https://blog.csdn.net/m0_50617544/article/details/129472724)

作者主页: [https://blog.csdn.net/m0\\_50617544](https://blog.csdn.net/m0_50617544)

```

118     int i, N;119     cin>>N;
120     nod nd, *rot=NULL;
121     for(i = 0; i < N; i++) scanf("%d", &in[i]);
122     for(i = 0; i < N; i++) scanf("%d", &pot[i]);
123     rot = recursion(rot, N-1, 0, N-1);
124     bool symmetric = width_first_symmetric(rot);
125     depth_first(rot);
126     bool treelove = judge_treelove();
127     if(treelove==true && symmetric==true) printf("Yes\n");
128     else printf("No\n");
129     for(i = 0; i < vec.size(); i++) printf("%d%s", vec[i], i==(vec.size()-1)?" ":" ");
130     return 0;
131 }

```

GitHub - ZouJiu1/PAT: PAT浙江大学题目解答内容

2023pat甲级春季考试最后一题\_SHSEDG的博客-CSDN博客

内容来源: csdn.net

作者昵称: 九是否随机的称呼

原文链接: [https://blog.csdn.net/m0\\_50617544/article/details/129472724](https://blog.csdn.net/m0_50617544/article/details/129472724)

作者主页: [https://blog.csdn.net/m0\\_50617544](https://blog.csdn.net/m0_50617544)