

|  |  |
| --- | --- |
|  |  |
| **JIANGSU UNIVERSITY** |

数据结构作业

2023-2024-1

学 号： 3220604077

姓 名： 唐熙晟

班 级： 信息安全2203

2023年 10 月 15 日

# 题目1

## 1 题目描述

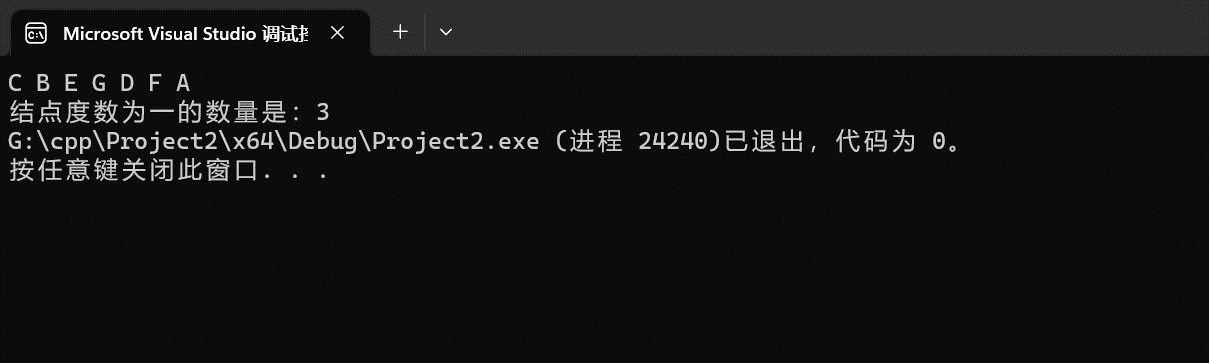
编写递归算法，求二叉树中度为1的结点的数量

## 2 系统功能

先序创建二叉树，层序寻找节点数

## 3运行界面截图与说明

测试样例：ABC@@DE@G@@F@@@



## 4 代码

1. #include<bits/stdc++.h>
2. using namespace std;
3. struct TreeNode {
4. string data;
5. TreeNode\* left;
6. TreeNode\* right;
7. };
8. TreeNode\* BuildTree(const string& str,int& index) {
9. if (index >= str.size() || str[index] == '@') {
10. return nullptr;
11. }
12. TreeNode\* tree = new TreeNode;
13. tree->data = (1, str[index]);
14. tree->left = nullptr;
15. tree->right = nullptr;
16. ++index;
17. tree->left = BuildTree(str, index);
18. ++index;
19. tree->right = BuildTree(str, index);
20. return tree;
21. }
22. TreeNode\* BuildTree(const string& str) {
23. int index = 0;
24. return BuildTree(str, index);
25. }
26. void InorderTraversal(TreeNode\* tree) {
27. if (tree == nullptr) {
28. return;
29. }
30. TreeNode\* curr = tree;
31. stack<TreeNode\*> dt;
32. while (curr != nullptr || !dt.empty()) {
33. while (curr != nullptr) {
34. dt.push(curr);
35. curr = curr->left;
36. }
37. curr = dt.top();
38. dt.pop();
39. cout << curr->data << " ";
40. curr = curr->right;
41. }
42. }
43. int findnumchildren(TreeNode\* tree) {
44. if (tree == nullptr) {
45. return 0;
46. }
47. queue<TreeNode\*> que;
48. que.push(tree);
49. int numchildren = 0;
50. while (!que.empty()) {
51. TreeNode\* curr = que.front();
52. que.pop();
53. if (curr->left == nullptr && curr->right == nullptr) {
54. numchildren += 1;
55. }
56. if(curr->left)
57. que.push(curr->left);
58. if(curr->right)
59. que.push(curr->right);
60. }
61. return numchildren;
62. }
63. int main() {
64. string s = "ABC@@DE@G@@F@@@";
65. TreeNode\* tree = BuildTree(s);
66. InorderTraversal(tree);
67. cout << endl;
68. cout << "结点度数为一的数量是：" << findnumchildren(tree);
69. }