《数据结构》课程实验报告

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 专业名称 | 计算机科学与技术（国际联合培养） | 年级 | 大二 | 班级 | 计联培1901 |
| 学生姓名 | 姬彬荃 | 学号 | 201911020125 | 指导教师 | 魏晓超 |
| 实验题目 | 实验八：BFS&DFS | | | 提交时间 | 2020.11.18 |

一、实验目的和要求

1、通过DFS算法实现对图的遍历

2、通过BFS算法实现对图的遍历

二、实验环境

Dev C++

三、实验内容及实施

（包括设计思想、源程序、参数说明、采用的关键理论、技术、算法或数据结构等的说明。

**【源程序】**



1. /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
2. \*\* 功能 : 数据结构实验八：图的DFS与BFS遍历
3. \*\* 作者 : 2019Jibinquan
4. \*\* 创建 : 2020-11-18 / 15:51
5. \*\* 联系 : 431263064@qq.com
6. /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
7. #include <iostream>
8. #include <cstdio>
9. #include <cmath>
10. #include <algorithm>
11. #include <cstring>
12. #include <queue>
13. #include <stack>
14. #include <vector>
15. #include <map>
16. #include <set>
17. #define ios ios::sync\_with\_stdio(false);cin.tie(0);cout.tie(0);
18. #define debug(a) cout << #a << " " << a << endl
19. #define inr register int
20. **using** **namespace** std;
21. **typedef** **long** **long** ll;
22. **const** **double** pi = acos(-1);
23. **const** **double** eps = 1e-8;
24. **const** **int** inf = 0x3f3f3f3f;
25. **const** **int** maxn = 100007;//1e5+7
26. **const** ll mod = 1000000007;//1e9+7
28. vector<**int**>G[maxn];
30. **int** n, m, len = 0;
31. **int** vis[maxn];
32. **int** dans[maxn];
33. **int** bans[maxn];

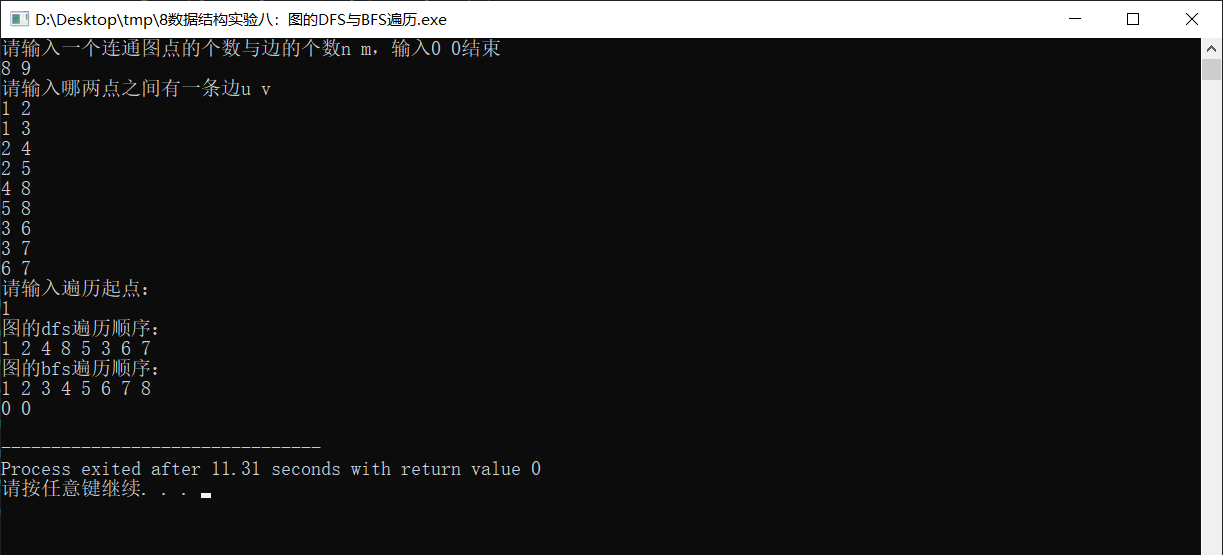
36. /\*----------------------------------\*
37. Function Definition
38. \*-----------------------------------\*/

41. **void** add\_edge(**int** u, **int** v)
42. {
43. G[u].push\_back(v);
44. G[v].push\_back(u);
45. }
47. **void** init(**int** st)
48. {
49. len = 0;
50. memset(vis, 0, **sizeof**(vis));
51. vis[st] = 1;
52. }
54. **void** dfs(**int** st)
55. {
56. dans[++len] = st;
57. **for** (**int** i = 0; i < G[st].size(); i++) {
58. **if** (!vis[G[st][i]]) {
59. vis[G[st][i]] = 1;
60. dfs(G[st][i]);
61. }
62. }
63. }
65. **void** bfs(**int** st)
66. {
67. queue<**int**>q;
68. q.push(st);
69. **while** (!q.empty()) {
70. **int** nw = q.front();
71. bans[++len] = nw;
72. q.pop();
73. **for** (**int** i = 0; i < G[nw].size(); i++) {
74. **if** (!vis[G[nw][i]]) {
75. vis[G[nw][i]] = 1;
76. q.push(G[nw][i]);
77. }
78. }
79. }
80. }

83. /\*----------------------------------\*
84. Main Function
85. \*-----------------------------------\*/

88. **int** main()
89. {
90. cout << "请输入一个连通图点的个数与边的个数n m，输入0 0结束" << endl;
91. **while** (cin >> n >> m) {
92. **if** (!(n || m)) {
93. **break**;
94. }
95. **else** {
96. cout << "请输入哪两点之间有一条边u v" << endl;
97. **for** (**int** i = 1, u, v; i <= m; i++) {
98. cin >> u >> v;
99. add\_edge(u, v);
100. }
101. **int** st;
102. cout << "请输入遍历起点：" << endl;
103. cin >> st;
104. init(st);
105. dfs(st);
106. init(st);
107. bfs(st);
108. cout << "图的dfs遍历顺序：" << endl;
109. **for** (**int** i = 1; i <= n; i++) {
110. cout << dans[i] << " \n"[i == n];
111. }
112. cout << "图的bfs遍历顺序：" << endl;
113. **for** (**int** i = 1; i <= n; i++) {
114. cout << bans[i] << " \n"[i == n];
115. }
116. }
117. }
118. //system("pause");
119. **return** 0;
120. }

**四、实验结果 (程序的执行结果)**



**五、实验讨论（可选）**

暂无