

**DFS：**

1. #include<iostream>
2. **using** **namespace** std;
4. **int** a[11][11];
5. **bool** visited[11];
7. **void** store\_graph()  //邻接矩阵存储图
8. {
9. **int** i,j;
11. **for**(i=1;i<=10;i++)
12. **for**(j=1;j<=10;j++)
13. cin>>a[i][j];
14. }
16. **void** dfs\_graph()    //深度遍历图
17. {
18. **void** dfs(**int** v);
20. memset(visited,**false**,**sizeof**(visited));
22. **for**(**int** i=1;i<=10;i++)  //遍历每个顶点是为了防止图不连通时无法访问每个顶点
23. **if**(visited[i]==**false**)
24. dfs(i);
25. }
27. **void** dfs(**int** v)  //深度遍历顶点
28. {
29. **int** Adj(**int** x);
31. cout<<v<<" ";  //访问顶点v
32. visited[v]=**true**;
34. **int** adj=Adj(v);
35. **while**(adj!=0)
36. {
37. **if**(visited[adj]==**false**)
38. dfs(adj);      //递归调用是实现深度遍历的关键所在
40. adj=Adj(v);
41. }
42. }
44. **int** Adj(**int** x)   //求邻接点
45. {
46. **for**(**int** i=1;i<=10;i++)
47. **if**(a[x][i]==1 && visited[i]==**false**)
48. **return** i;
50. **return** 0;
51. }
53. **int** main()
54. {
55. cout<<"初始化图:"<<endl;
56. store\_graph();
58. cout<<"dfs遍历结果:"<<endl;
59. dfs\_graph();
61. **return** 0;
62. }

**BFS：**

1. #include<iostream>
2. #include<queue>
3. **using** **namespace** std;
5. **int** a[11][11];
6. **bool** visited[11];
8. **void** store\_graph()
9. {
10. **for**(**int** i=1;i<=10;i++)
11. **for**(**int** j=1;j<=10;j++)
12. cin>>a[i][j];
13. }
15. **void** bfs\_graph()
16. {
17. **void** bfs(**int** v);
19. memset(visited,**false**,**sizeof**(visited));
21. **for**(**int** i=1;i<=10;i++)
22. **if**(visited[i]==**false**)
23. bfs(i);
24. }
26. **void** bfs(**int** v)
27. {
28. **int** Adj(**int** x);
30. queue<**int**> myqueue;
31. **int** adj,temp;
33. cout<<v<<" ";
34. visited[v]=**true**;
35. myqueue.push(v);
37. **while**(!myqueue.empty())    //队列非空表示还有顶点未遍历到
38. {
39. temp=myqueue.front();  //获得队列头元素
40. myqueue.pop();         //头元素出对
42. adj=Adj(temp);
43. **while**(adj!=0)
44. {
45. **if**(visited[adj]==**false**)
46. {
47. cout<<adj<<" ";
48. visited[adj]=**true**;
49. myqueue.push(adj);   //进对
50. }
52. adj=Adj(temp);
53. }
54. }
55. }
57. **int** Adj(**int** x)
58. {
59. **for**(**int** i=1;i<=10;i++)
60. **if**(a[x][i]==1 && visited[i]==**false**)
61. **return** i;
63. **return** 0;
64. }
66. **int** main()
67. {
68. cout<<"初始化图:"<<endl;
69. store\_graph();
71. cout<<"bfs遍历结果:"<<endl;
72. bfs\_graph();
74. **return** 0;
75. }