



DFS:

```

1. #include<iostream>
2. using namespace std;
3.
4. int a[11][11];
5. bool visited[11];
6.
7. void store_graph() //邻接矩阵存储图
8. {
9.     int i,j;
10.
11.     for(i=1;i<=10;i++)
12.         for(j=1;j<=10;j++)
13.             cin>>a[i][j];
14. }
15.
16. void dfs_graph() //深度遍历图
17. {
18.     void dfs(int v);
19.
20.     memset(visited,false,sizeof(visited));
21.
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. }
26.

```

```

27. void dfs(int v) //深度遍历顶点
28. {
29.     int Adj(int x);
30.
31.     cout<<v<<" "; //访问顶点 v
32.     visited[v]=true;
33.
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
39.
40.         adj=Adj(v);
41.     }
42. }
43.
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;
49.
50.     return 0;
51. }
52.
53. int main()
54. {
55.     cout<<"初始化图:"<<endl;
56.     store_graph();
57.
58.     cout<<"dfs 遍历结果:"<<endl;
59.     dfs_graph();
60.
61.     return 0;
62. }

```

BFS:

```

1. #include<iostream>
2. #include<queue>
3. using namespace std;
4.
5. int a[11][11];
6. bool visited[11];

```

```

7.
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. }
14.
15. void bfs_graph()
16. {
17.     void bfs(int v);
18.
19.     memset(visited,false,sizeof(visited));
20.
21.     for(int i=1;i<=10;i++)
22.         if(visited[i]==false)
23.             bfs(i);
24. }
25.
26. void bfs(int v)
27. {
28.     int Adj(int x);
29.
30.     queue<int> myqueue;
31.     int adj,temp;
32.
33.     cout<<v<<" ";
34.     visited[v]=true;
35.     myqueue.push(v);
36.
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();        //头元素出队
41.
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); //进队
50.             }

```

```
51.
52.         adj=Adj(temp);
53.     }
54. }
55. }
56.
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;
62.
63.     return 0;
64. }
65.
66. int main()
67. {
68.     cout<<"初始化图:"<<endl;
69.     store_graph();
70.
71.     cout<<"bfs 遍历结果:"<<endl;
72.     bfs_graph();
73.
74.     return 0;
75. }
```