

广度优先搜索和深度优先搜索 (c++)

原创

渴望成长的旅行鼠

已于 2023-05-09 00:41:37 修改

阅读量1.2k

收藏 17

点赞数 1

文章标签: 深度优先 宽度优先 c++ 算法

版权

广度优先搜索 和深度优先搜索是两种常见的图搜索算法。

广度优先搜索（BFS）：从起点开始，按照距离逐层向外搜索，直到找到目标节点或搜索完所有节点。BFS一般使用队列来实现。

深度优先搜索（DFS）：从起点开始，沿着一条路径一直搜索到底，直到找到目标节点或者无法继续搜索为止。DFS一般使用递归或栈来实现。

下面是使用C++实现BFS和DFS的示例代码：

BFS实现:

```c++

```
1 #include <iostream>
2 #include <queue>
3 #include <vector>
4
5 using namespace std;
6
7 vector<int> BFS(vector<vector<int>>& graph, int start) {
8     vector<int> res;
9     queue<int> q;
10    vector<bool> visited(graph.size(), false);
11    q.push(start);
12    visited[start] = true;
13    while (!q.empty()) {
14        int cur = q.front();
15        q.pop();
16        res.push_back(cur);
17        for (int i : graph[cur]) {
18            if (!visited[i]) {
19                q.push(i);
20                visited[i] = true;
21            }
22        }
23    }
24    return res;
25 }
26
27 int main() {
28     vector<vector<int>> graph = {{1, 2}, {0, 2, 3}, {0, 1, 4}, {1, 4}, {2, 3}};
29     vector<int> res = BFS(graph, 0);
30     for (int i : res) {
31         cout << i << " ";
32     }
33     cout << endl;
34     return 0;
35 }
```

...

DFS实现:

```c++

```
1 #include <iostream>
2 #include <vector>
3
4 using namespace std;
5
6 void DFS(vector<vector<int>>& graph, int cur, vector<bool>& visited, vector<int>& res) {
7     visited[cur] = true;
8     res.push_back(cur);
9     for (int i : graph[cur]) {
10        if (!visited[i]) {
11            DFS(graph, i, visited, res);
12        }
13    }
14 }
15
16 vector<int> DFS(vector<vector<int>>& graph, int start) {
17     vector<int> res;
18     vector<bool> visited(graph.size(), false);
19     DFS(graph, start, visited, res);
20     return res;
21 }
22
23 int main() {
24     vector<vector<int>> graph = {{1, 2}, {0, 2, 3}, {0, 1, 4}, {1, 4}, {2, 3}};
25     vector<int> res = DFS(graph, 0);
26     for (int i : res) {
27         cout << i << " ";
28     }
29     cout << endl;
30     return 0;
31 }
```



渴望成长的旅行鼠

关注

1



17

1

