



算法分析 | 一文理解搜索算法-彻底入门DFS



汤包

欢迎订阅微信公众号：挑战算法与程序设计

关注他

50 人赞同了该文章

前言

搜索是一种适用性非常广泛的算法。很多我们实际遇到的问题，数据量可能不会太大，都可以暴力搜索一把来解决。

▲ 赞同 50 ▼

● 4 条评论

➦ 分享

♥ 喜欢

★ 收藏

📄 申请转载

...



其实这个算法网上也很多但是都讲的比较复杂很多人也只能看一个前面的东西后面的其实看不懂。也有很多文章讲的不是太精炼，讲不出个所以然来。所以小编今天就利用这篇文章来带大家先入门搜索算法。最简单的 DFS 相当于一个暴力搜索算法。

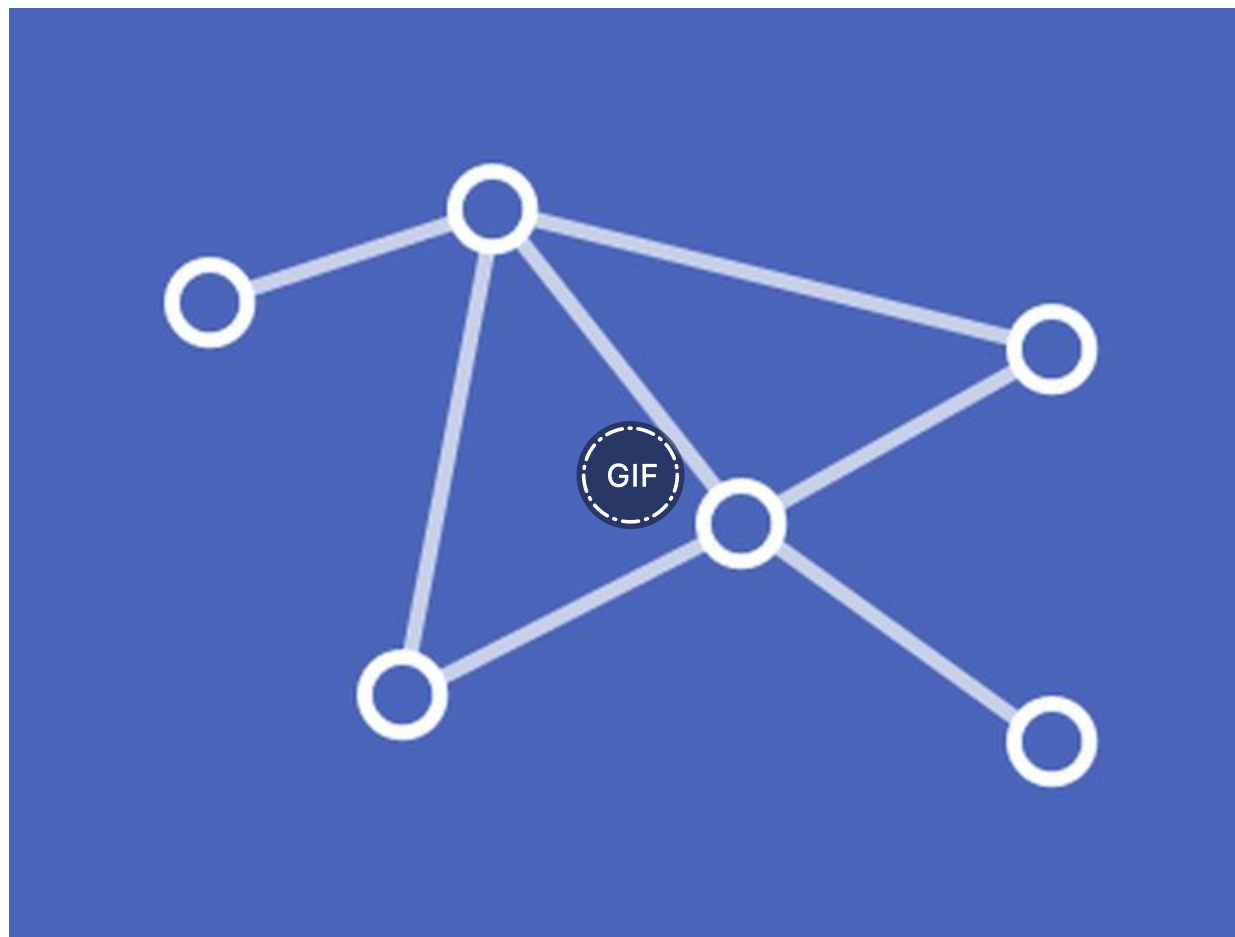
如果不了解暴力的可以看看前面写的这篇文章：

[算法分析 | 枚举+优化套路入门，万能的暴力法](#)

基本上遇到的很多题目暴力都能解决，毕竟 CPU 处理的速度这么快。

搜索DFS正文

搜索算法是属于一种比较基础的算法，相当于万丈高楼的第一层，也是后期学习的一些高级算法的基础部分,搜索算法分为深度优先搜索（Depth First Search, DFS）和广度优先搜索（Breadth First Search, BFS）这两种。DFS 相对简单一点那就从 DFS 开始入门吧。



说到这个搜索算法就要讲到图的问题，有学过离散数学或者数据结构的都知道，图是一个点集合加上一个边的集合一个边对应两个定点，属于一种二元关系。

如果你没学过的话上面那句话省略，直接看这里：举个例子讲，微信朋友圈可以看成一个很大的图，每个人都是一个顶点，彼此是好友彼此的朋友圈就能看到，相当于彼此的可达的。这个问题——抽象！

▲ 赞同 50 ▼

● 4 条评论

🔗 分享

❤️ 喜欢

★ 收藏

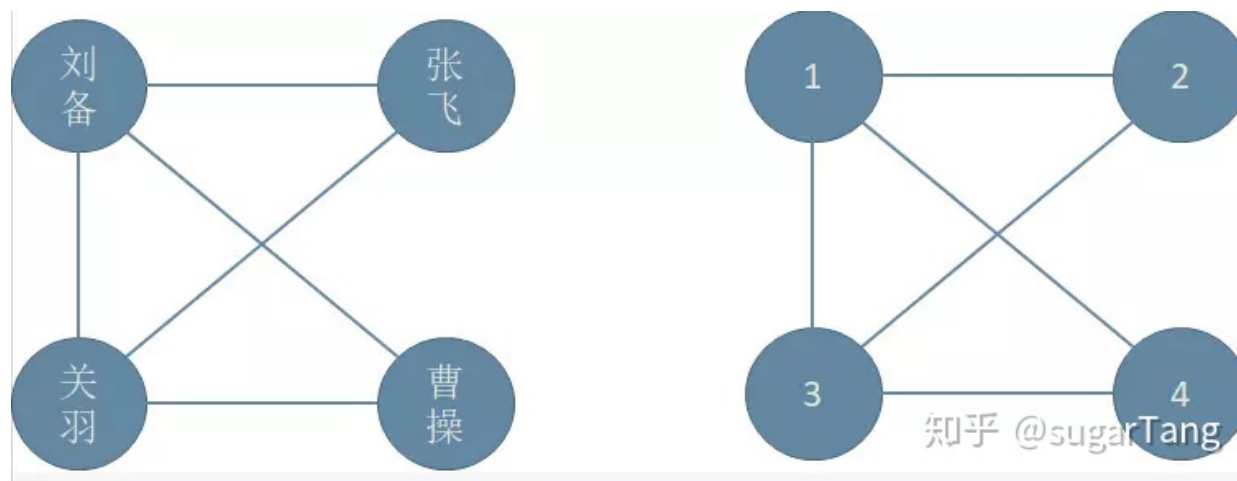
📄 申请转载

...

好友的好友也都转发.....然后全世界都知道你长什么样子了。所以小编的文章能怎么样还是需要你们这些第一批种子用户的。



直接形象化的上图



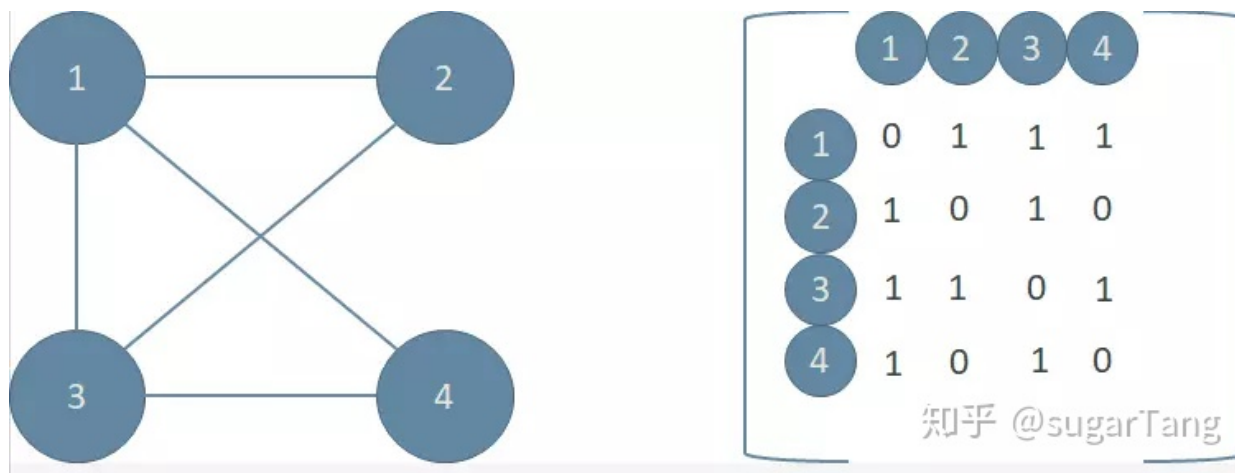
了解什么是图了，那么我们从图的遍历开始讲，给定一个包含N个顶点的图，以及图上的M条边。让你遍历图中的每一个顶点恰好一次。图的遍历有很多用途，比如判断一个图是不是连通的。和判断你的微信能不能连接全世界是一样的。

我们说到一个算法首先考虑的是他的实现，要实现的话就得要存储，看问题我们是两个集合（点集和边集）集合一般的思想就是通过一个数组集合来存储，通过对点集的标号，找到与数组下标的关系，我们就能实现存储。

其实1 ▲ 赞同 50 ▼ ● 4 条评论 ➦ 分享 ♥ 喜欢 ★ 收藏 📄 申请转载 ...



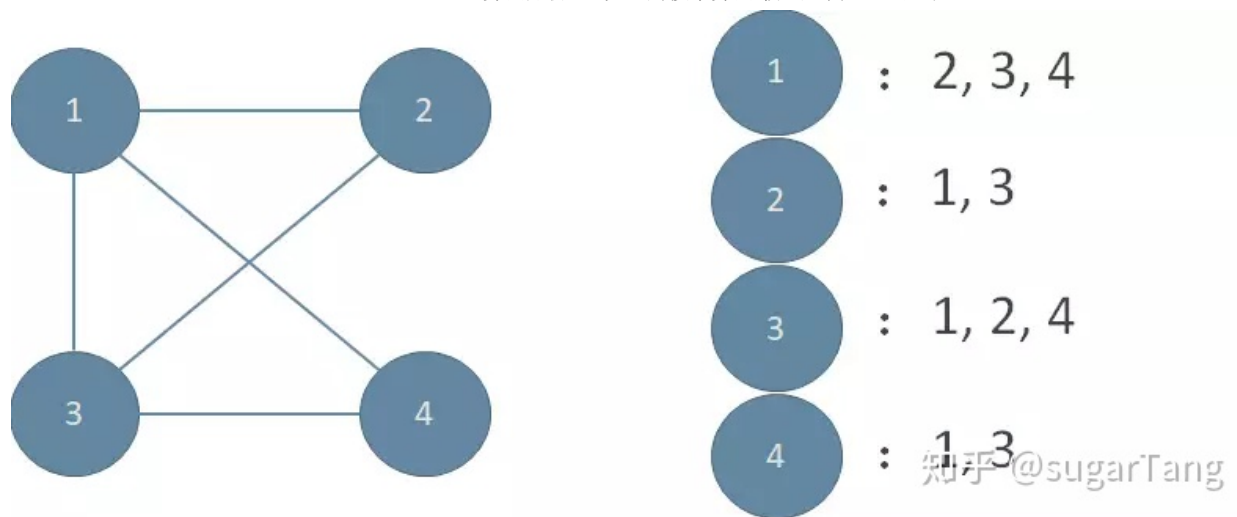
邻接矩阵是说我们用一个二维矩阵A来表示边集。 $A_{ij}=0$ 表示顶点i和顶点j之间不存在边， $A_{ij}=1$ 表示顶点i和顶点j之间存在边。如果我们用邻接矩阵表示上图，是这个样子：



在程序中，我们一般用一个二维数组来表示邻接矩阵：`int a[N+1][N+1]`。 $a[i][j]=1$ 表示i和j之间有边相连， $a[i][j]=0$ 表示i和j之间没有边相连。

邻接表是图中的每一个顶点i都有一个线性表，保存与i相连的顶点编号。

如果我们用邻接矩阵表示上图，是这个样子：



在程序中，我们一般用一个数组嵌套vector的方法来表示邻接表：`vector<int> g[N+1]`，里面保存着每一个与i相邻的顶点编号。

上面讲了什么是搜索，如何存储一个图。那么重头戏来了我们不妨设从1号顶点起始。在搜索过程中，我们维护一个布尔数组`bool visited[N+1]`，这个数组用来表示每个顶点是不是已经遍历过了。`visited[i]==true`表示顶点i已经遍历过了，`visited[i]==false`表示i还没有遍历过。DFS一般我们可以用递归实现，如果采用邻接矩阵，伪代码如下：



Visited[] = {FALSE, FALSE, ... FALSE}

DFS(x):

Visited[x] = TRUE

For i = 1 .. N:

If !Visited[i] AND A[x][i]:

DFS(i)

知乎 @sugarTang

当我们执行DFS(1)的时候，程序就会开始从1号节点开始遍历，每一次都对搜索过的书标记一直到全部被标记完为止。而每一次DFS执行中都要i循环从1到N遍历一遍。所以整个复杂度是 $O(N^2)$ 的。

了解了大概，看官们是不跃跃欲试了呢。那就看看我下面的栗子，**源码的获取方式在文章末尾**。小编编写了两种方法。

举个栗子



描述：

给定一个包含 N 个顶点的图，顶点编号 $1 \sim N$ 。请你判断这个图是不是连通的？

输入：

第一行包含两个整数 N 和 M ，代表顶点数和边数。 $(1 \leq N, M \leq 100000)$

以下 M 行每行包含两个整数 u 和 v ，代表 u 和 v 之间有一条边。 $(1 \leq u, v \leq N)$

输出：

如果图是连通的，输出 YES；否则输出 NO

知乎 @sugarTang

小提示：

这道题的基本思路就是从1号节点出发开始遍历。如果遍历结束时所有visited[]数组的值全都是true，那么就说明整个图是连通的，否则就不是。

C语言代码是这个样子：



```
1  /*
2  DFS-连接矩阵 一般方法 -C语言
3  by sugarTang
4  data: 2018-8-4
5  */
6  #include <stdio>
7  #include <algorithm>
8  using namespace std;
9  const int N = 10002;
10 bool vis[N];
11 int g[N][N];
12 int m,n;
13 int cnt=0;
14
15 void dfs(int x){
16     vis[x]=true;
17     cnt++;
18     for(int i=1; i<=n; i++){
19         if(!vis[i]) {
20             dfs(i);
21         }
22     }
23 }
24 int main()
25 {
26     scanf("%d%d",&n,&m);
27     for(int i=1; i<=n; i++) vis[i]=false;
28     for(int i=1; i<=m; i++){
29         int a,b;
30         scanf("%d%d",&a,&b);
31         g[a][b]=1;
32     }
33     dfs(1);
34     bool ans = true;
35     for(int i=1; i<=n; i++) ans = ans && vis[i];
36     if(ans){
37         printf("YES\n");
38     }
39     else {
40         printf("NO\n");
41     }
42 }
```

▲ 赞同 50 ▼

● 4 条评论

➦ 分享

♥ 喜欢

★ 收藏

📄 申请转载

...



C++代码使用STL：

▲ 赞同 50 ▼ ● 4 条评论 ➦ 分享 ❤ 喜欢 ★ 收藏 📄 申请转载 ...



```
1  /*
2  DFS-连接矩阵 C++ STL库
3  by sugarTang
4  data: 2018-8-4
5  */
6
7  #include <iostream>
8  #include <vector>
9  using namespace std;
10 const int N = 100002;
11 bool vis[N];
12 vector<int> g[N+1];
13 int n,m;
14
15
16 void dfs(int x){
17     vis[x] = true;
18     for(int i : g[x]){
19         // cout << i << endl;
20         if(!vis[i]) dfs(i);
21     }
22 }
23
24 int main()
25 {
26     cin >> n >> m;
27     for(int i=0; i<n; i++) vis[i]=false;
28     for(int i=0; i<m; i++){
29         int a,b;
30         cin >> a >> b;
31         g[a].push_back(b);
32         g[b].push_back(a);
33     }
34     dfs(1);
35     bool ans = true;
36     for(int i=1; i<=n; i++) ans = ans && vis[i];
37     if(ans) cout << "YES" << endl;
38     else cout << "NO" << endl;
39     return 0;
40 }
```

知乎 @sugarTang

最后一点，就是 DFS 运用的不同的场景限制会有些不同，有些时候需要去掉重复的路径，还有的是预防存在环路，代码死循环，或者没有找完的情况。如果有问题在后台留言。



代码获取

为了方便看官们的代码获取，小编在微信公众号后台放了获取方式。微信搜索：「挑战算法与程序设计」订阅此号，然后后台回复：**搜索01**即可获得两种方式的代码。

公众号出了这个还有更加精彩的哟~！、

编辑于 2018-08-05

[算法设计](#) [搜索算法](#) [算法竞赛入门经典（书籍）](#)

推荐阅读

Jump Point Search(JPS)算法总结与实现（附Demo）

关于这篇文章第一次翻阅A星算法的文章，是为了弄清楚A星在游戏开发中的地位。当是在脑海中没有形成它的算法模型，也苦于没有一个自定义轨迹且又能展示每一步的细节的demo，于是自己动

Thisl...

▲ 赞同 50



● 4 条评论

🔗 分享

♥ 喜欢

★ 收藏

📄 申请转载



干货 | 快速掌握禁忌搜索算法求解带时间窗的车辆路径问题

本文发布于公众号【数据魔术师】同名文章 — 什么是禁忌搜索算法？禁忌搜索算法（Tabu Search Algorithm，简称TS）起源于对于人类记忆功能的模仿，是一种 亚启发

常用算法指南(一)基本算法思想

常用算法指南(一)基本算法思想近期看了《Java常用算法手册》，来总结一下。想看原书的，在这里提供原书链接：《Java常用算法手册.pdf》前言编码不仅仅是把代码写出来，还要求能够清晰的表达...



算法总结-二分查找法



4 条评论

[切换为时间排序](#)

写下你的评论...



英雄登场

2020-04-08

可能是我无知，这个vector数组好舒服，我平时都用邻接链表[捂脸][捂脸][捂脸][捂脸]，比这个复杂多了

3



王大仙 回复 英雄登场

2020-12-29

这样的确好写了，但是用邻接矩阵做DFS比用邻接链表做DFS时间复杂度更高。邻接矩阵DFS时间复杂度与顶点数 n 有关，平均时间复杂度为 $O(n^2)$ 。邻接链表DFS时间复杂度与顶点数 n 和边数 e 有关，平均时间复杂度只有 $O(n+e)$ 。况且如果是稀疏图，邻接矩阵的空间浪费也很严重。

赞



roadme

03-31

可否别误人子弟，邻接矩阵这个条件都没用到，输入边的关系有鸟用

1



九月

2020-04-12

dev-c++好评!

赞

赞同 50



4 条评论

分享

喜欢

收藏

申请转载

