

六维空间
我的学习笔记本儿~~~

≤	2011年5月							≥
	日	一	二	三	四	五	六	
	1	2	3	4	5	6	7	
	8	9	10	11	12	13	14	
	15	16	17	18	19	20	21	
	22	23	24	25	26	27	28	
	29	30	31	1	2	3	4	
	5	6	7	8	9	10	11	

昵称: [geeker](#)
园龄: 9年3个月
粉丝: 79
关注: 11
[+加关注](#)

搜索

找找看

谷歌搜索

常用链接

- [我的随笔](#)
- [我的评论](#)
- [我的参与](#)
- [最新评论](#)
- [我的标签](#)

积分与排名

积分 - 38905
排名 - 23002

随笔分类

- [Android 初学笔记\(6\)](#)
- [BFS\(2\)](#)
- [DFS+回溯\(5\)](#)
- [HDOJ\(1\)](#)
- [Others\(1\)](#)
- [POJ\(11\)](#)
- [USACO\(5\)](#)
- [并查集\(7\)](#)
- [递归\(4\)](#)
- [动态规划\(16\)](#)
- [分布式计算\(1\)](#)
- [哈希表\(2\)](#)
- [枚举\(1\)](#)
- [模拟\(1\)](#)
- [树\(2\)](#)

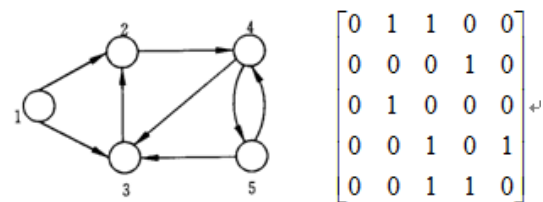
[博客园](#) [首页](#) [新随笔](#) [联系](#) [订阅](#) [RSS](#) [管理](#)

posts - 74, comments - 36, trackbacks - 0

算法与数据结构（2）——图的表示法与常用的转化算法

《图的表示方法》

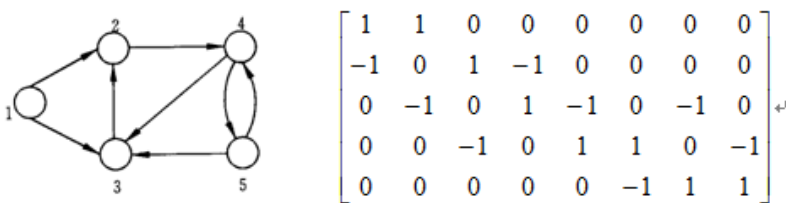
(i) 邻接矩阵表示法，如图：



也就是说，如果两节点之间有一条弧，则邻接矩阵中对应的元素为1；否则为0。可以看出，这种表示法非常简单、直接。但是，在邻接矩阵的所有 n^2 个元素中，只有 m 个为非零元。如果网络比较稀疏，这种表示法浪费大量的存储空间，从而增加了在网络中查找弧的时间。

同样，对于网络中的权，也可以用类似邻接矩阵的 矩阵表示。只是此时一条弧所对应的元素不再是1，而是相应的权而已。如果网络中每条弧赋有多种权，则可以用多个矩阵表示这些权。

(ii) 关联矩阵表示法



也就是说，在关联矩阵中，每行对应于图的一个节点，每列对应于图的一条弧。如果一个节点是一条弧的起点，则关联矩阵中对应的元素为1；如果一个节点是一条弧的终点，则关联矩阵中对应的元素为 -1；如果一个节点与一条弧不关联，则关联矩阵中对应的元素为0。对于简单图，关联矩阵每列只含有两个非零元（一个 1，一个 -1）可以看出，这种表示法也非常简单、直接。但是，在关联矩阵的所有 mn 个元素中，只有 $2m$ 个为非零元。如果网络比较稀疏，这种表示法也会浪费大量的存储空间。但由于关联矩阵有许多特别重要的理论性质，因此它在网络优化中是非常重要的概念。

同样，对于网络中的权，也可以通过对联接矩阵的扩展来表示。例如，如果网络中每条弧有一个权，我们可以把关联矩阵增加一行，把每一条弧所对应的权存储在增加的行中。如果网络中每条弧赋有多个权，我们可以把关联矩阵增加相应的行数，把每一条弧所对应的权存储在增加的行中。

(iii) 弧表示法

数据结构(18)
 数学 (组合, 数论) (3)
 算法学习(12)
 他人报告收藏(2)
 贪心(4)
 王晓东 算法教材学习笔记(4)

随笔档案

2012年6月(6)
 2012年5月(1)
 2011年10月(16)
 2011年9月(9)
 2011年7月(3)
 2011年5月(26)
 2011年4月(13)

最新评论

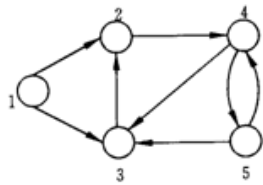
1. Re:JAVA零基础——坦克大战 (上) ...
 @ geeker 下一篇在哪? ...
 --linlin★
 2. Re:USACO 2.3.4
 感觉你的博客好多知识啊。你是搞过ACM的队员吗?
 --meetviolet
 3. Re:归并排序(递归实现+非递归实现+自然合并排序)
 void mergeSort2(int n){ int s=2,i; while(s<=n){ i=0; while(i+s<=n){ merge(i,i+s-1,i+s/2-1); i+=s; } ...
 --gxiaob
 4. Re:算法与数据结构 (2) ——图的表示法与常用的转化算法
 前向星还是没怎么明白
 --HPU-张朋飞
 5. Re:POJ-1141-完全加括号-DP
 错代码你也发, 受不了你
 --lishimin_come

阅读排行榜

1. 动态规划——矩阵连乘的问题(30545)
 2. 归并排序(递归实现+非递归实现+自然合并排序)(8941)
 3. 算法与数据结构 (2) ——图的表示法与常用的转化算法(7124)
 4. JAVA零基础——坦克大战 (上) (4871)
 5. 数据结构——堆排序(4273)

评论排行榜

1. POJ 2018 动态规划(7)
 2. 记忆化搜索 (附例题HDOJ 1501) (7)
 3. JAVA零基础——坦克大战 (上) (7)
 4. 动态规划——矩阵连乘的问题(3)
 5. 数据结构——堆排序(2)



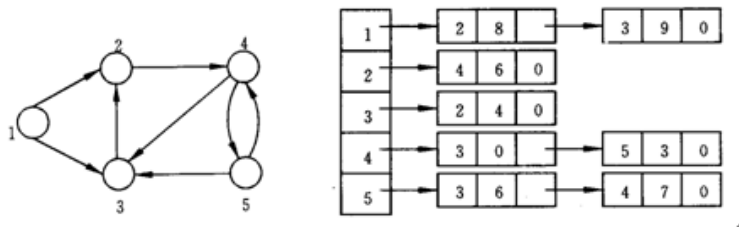
起点	1	1	2	3	4	4	5	5
终点	2	3	4	2	3	5	3	4
权	8	9	6	4	0	3	6	7

例如, 例7所示的图, 假设弧(1,2), (1,3), (2,4), (3,2), (4,3), (4,5), (5,3)和(5,4)上的权分别为8, 9, 6, 4, 0, 3, 6和7, 则弧表表示如上:

为了便于检索, 一般按照起点、终点的字典序顺序存储弧表, 如上面的弧表就是按照这样的顺序存储的。

(iv) 邻接表表示法

邻接表表示法将图以邻接表 (adjacency lists) 的形式存储在计算机中。所谓图的邻接表, 也就是图的所有节点的邻接表的集合; 而对每个节点, 它的邻接表就是它的所有出弧。邻接表表示法就是对图的每个节点, 用一个单向链表列出从该节点出发的所有弧, 链表中每个单元对应于一条出弧。为了记录弧上的权, 链表中每个单元除列出弧的另一个端点外, 还可以包含弧上的权等作为数据域。图的整个邻接表可以用一个指针数组表示。例如, 例7所示的图, 邻接表表示为



(v) 星形表示法

星形 (star) 表示法的思想与邻接表表示法的思想有一定的相似之处。对每个节点, 它也是记录从该节点出发的所有弧, 但它不是采用单向链表而是采用一个单一的数组表示。也就是说, 在该数组中首先存放从节点1出发的所有弧, 然后接着存放从节点2出发的所有弧, 依此类推, 最后存放从节点 出发的所有弧。对每条弧, 要依次存放其起点、终点、权的数值等有关信息。这实际上相当于对所有弧给出了一个顺序和编号, 只是从同一节点出发的弧的顺序可以任意排列。此外, 为了能够快速检索从每个节点出发的所有弧, 我们一般还用数组记录每个节点出发的弧的起始地址 (即弧的编号)。在这种表示法中, 可以快速检索从每个节点出发的所有弧, 这种星形表示法称为前向星形 (forward star) 表示法。

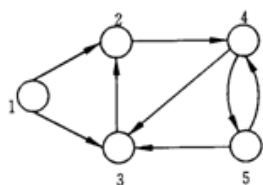
例如, 在例7所示的图中, 仍然假设弧 (1,2), (1,3), (2,4), (3,2), (4,3), (4,5), (5,3) 和 (5,4) 上的权分别为8, 9, 6, 4, 0, 3, 6和7。此时该网络图可以用前向星形表示法表示如下:

节点对应的出弧的起始地址编号数组 (记为 *point*)

节点号 <i>i</i>	1	2	3	4	5	6
起始地址 <i>point(i)</i>	1	3	4	6	7	9

记录弧信息的数组

弧编号	1	2	3	4	5	6	7	8
起点	1	1	2	3	4	4	5	5
终点	2	3	4	2	3	5	3	4
权	8	9	6	4	0	3	6	7



《星形表示法详解及其转化算法》

推荐排行榜

1. 算法与数据结构 (2) ——图的表示法与常用的转化算法 (7)
2. JAVA零基础——坦克大战 (上) (5)
3. 动态规划——矩阵连乘的问题(3)
4. 记忆化搜索 (附例题HDOJ 1501) (3)
5. 归并排序(递归实现+非递归实现+自然合并排序)(2)

注意：上面的第一张表实际上有个错误，仔细看的童鞋应该能发现，起始地址point(i) : 1, 3, 4, 5, 7, 9, 那个6应该是5

通常会设置一个st[i] 数组，和STL类似，[st[i], st[i+1]) 恰好为以结点i开头的边下标。对应于上个例子的第一张表，则该数组为：

```
st[6]={1,3,4,5,7,9};
```

还会有一个数组对应于第二张表，主要使用第三行数据，

```
v[8]={2,3,4,2,3,5,3,4};
```

下面的程序把树的前向星表示转化成左儿子-右兄弟表示，以方便后续算法实现。

```

1 void star2lsrs ()
2 {
3     memset (son , 0 , sizeof (son )); /*清零， 为零代表链表为空son */
4     for(i = 1; i <= n; i ++ )
5         /*按逆序考虑各个结点，则最后的链表是顺序的*/
6         for(j = st[i + 1] - 1; j >= st[i ]; j -- )
7             {
8                 bro[j ] = son[i];
9                 son[i ] = v[j ]; /*插到链表首部*/
10            }
11 }
```

图最常用的表示法是邻接矩阵和邻接表。对于静态图（建图完毕后不再修改图的结构）往往用前向星来代替邻接表，节省空间和时间。

邻接矩阵不管输入格式如何，总是很容易得到邻接矩阵，只需要注意平行边的情况。前向星邻接矩阵本身就包含了顶点序，因此很容易转化为前向星：

把邻接矩阵转换为前向星表示法：

```

1 void matrix2star ()
2 {
3     /*上一条的第一端点初始化为（表示未出现），边数初始化为u0m0 */
4     u = m = 0;
5     for(i = 1; i <= n; i ++ )
6         for(j = 1; j <= n; j ++ )
7             {
8                 if(a[i][j])
9                     {
10                        v[++m] = j;
11                        while (u < i)
12                            st [++u] = m;
13                    }
14            }
15 }
16 /*
17 在程序中，u代表上一条边的第一个顶点编号，当u < i时代表这条边的第一端点还没有出现过，设置st[u + 1] : : : st[i]为m。
18 */
19 */
```

把边列表转化成前向星的方法类似，只需要把第一顶点相同的结点串成链表，用计数器法进行结点编号分配，和前向星转化成左儿子-右兄弟一样每次插入到链表首部，在O(m)时间内可以建立前向星表示。当然，也可以按第一顶点为关键字直接进行快速排序，不过速度稍微慢一些

分类: [数据结构](#)

[好文要顶](#)
[关注我](#)
[收藏该文](#)




geeker

关注 - 11

粉丝 - 79

[+加关注](#)

7

1

« 上一篇: [算法与数据结构（1）——栈和队列初步认识](#)» 下一篇: [数据结构与算法（3）——二叉堆](#)posted on 2011-05-06 21:11 [geeker](#) 阅读(7151) 评论(2) [编辑](#) [收藏](#)**FeedBack:**

#1楼

2011-05-08 08:22 | [Tanky_Woo](#)

不错，支持一个。

支持(0) 反对(0)

#2楼

2013-05-24 08:12 | [火星十一郎](#)

前向星还是没怎么明白

支持(0) 反对(0)

[刷新评论](#) [刷新页面](#) [返回顶部](#)**注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#)，[访问](#) [网站首页](#)。**[【推荐】超50万行VC++源码: 大型组态工控、电力仿真CAD与GIS源码库](#)[【推荐】了不起的开发者，挡不住的华为，园子里的品牌专区](#)[【推荐】《Java开发手册》测试题大闯关，看看你能走到哪一步](#)[【推荐】有道智云周年庆，API服务大放送，注册即送100元体验金！](#)[【推荐】技术人必备的17组成长笔记+1500道面试题](#)**相关博文:**

- [数据结构与算法（六），图](#)
 - [浅析数据结构-图的基本概念](#)
 - [图的存储结构\(邻接矩阵与邻接表\)及其C++实现](#)
 - [数据结构基础温故-5.图（中）：最小生成树算法](#)
 - [浅谈算法和数据结构: 十二 无向图相关算法基础](#)
- » [更多推荐...](#)

最新 IT 新闻:

- [中国市场智能机“变天”了：联发科处理器占有率超高通、重回一哥](#)
 - [佳兆业国际集团牵手腾讯云，开启全面数字化转型](#)
 - [抢救TikTok们：这场保卫战才刚刚开始](#)
 - [强买强卖还要分成 白嫖之王特朗普](#)
 - [落寞的P2P、退场的共享单车，新经济为何玩着玩着就死了？](#)
- » [更多新闻...](#)

