

**目录**

[一、图的广度优先遍历 1](#_Toc511939307)

[2.1广度优先搜索介绍 1](#_Toc511939308)

[2.2有向图的广度优先遍历 1](#_Toc511939309)

[二、爬虫算法 2](#_Toc511939310)

[2.1互联网的抽象 2](#_Toc511939311)

[2.2去重策略 2](#_Toc511939312)

[2.3爬虫算法 2](#_Toc511939313)

[三、参考文献 3](#_Toc511939314)

# 一、图的广度优先遍历

## 2.1广度优先搜索介绍

广度优先搜索算法(Breadth First Search)，又称为"宽度优先搜索"或"横向优先搜索"，简称BFS。

它的思想是：从图中某顶点v出发，在访问了v之后依次访问v的各个未曾访问过的邻接点，然后分别从这些邻接点出发依次访问它们的邻接点，并使得“先被访问的顶点的邻接点先于后被访问的顶点的邻接点被访问，直至图中所有已被访问的顶点的邻接点都被访问到。如果此时图中尚有顶点未被访问，则需要另选一个未曾被访问过的顶点作为新的起始点，重复上述过程，直至图中所有顶点都被访问到为止。

换句话说，广度优先搜索遍历图的过程是以v为起点，由近至远，依次访问和v有路径相通且路径长度为1,2...的顶点。

## 2.2有向图的广度优先遍历

下面以图2-1为例，进行说明

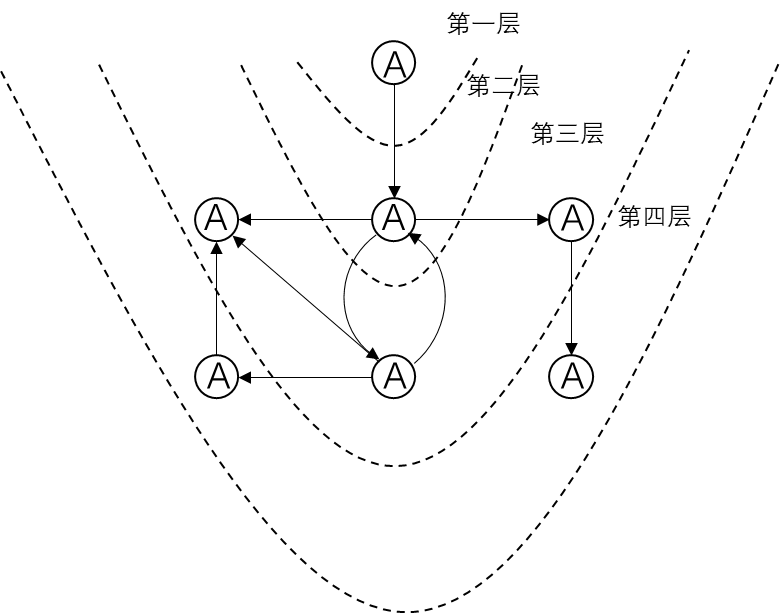


图2-1

第1步：访问A。

第2步：访问B。

第3步：依次访问C,E,F。

在访问了B之后，接下来访问B的出边的另一个顶点，即C,E,F。前面已经说过，在本文实现中，顶点ABCDEFG按照顺序存储的，因此会先访问C，再依次访问E,F。

第4步：依次访问D,G。

在访问完C,E,F之后，再依次访问它们的出边的另一个顶点。还是按照C,E,F的顺序访问，C的已经全部访问过了，那么就只剩下E,F；先访问E的邻接点D，再访问F的邻接点G。

因此访问顺序是：A -> B -> C -> E -> F -> D -> G

# 二、爬虫算法

## 2.1互联网的抽象

我们将整个互联网抽象成一张有向连接图，网页抽象为定点，定点之间的连接即为url之间的相互跳转。

## 2.2去重策略

一个原始的想法就是将所有已访问过的所有url进行顺序存储。你可以把全部已经下载完成的url存储到数据库中。每次有一个爬虫线程得到一个任务url开始下载之前，到数据库中检索此url，如果没有出现过，则将这个此url写入数据库。

但是这种去重的的思想是非常直观的。但是占用存储空间不说，查找效率超级低下，因此这个方案行不通。

对url进行hash运算映射到某个地址，将该url 和hash值当做键值对存放到hash表中，只需要对需要检测的URL的hash的映射进行比对，从而就可以对url是否存在进行判断。因此，原来的url库就可以简化为hash库，这要比url简便很多，但是需要考虑 hash 碰撞的问题，在设计中需要对hash函数进行考虑，避免因考虑不周造成hash碰撞。

所以我们采用使用url的MD5码去重的方法，MD5码基于hash算法，MD5算法能够将任何字符串压缩为128位整数，并映射为物理地址，同时，与传统的Hash去重方法相比，MD5进行Hash映射碰撞概率很低。MD5经过时间验证，是一种比较好的去重方法。

## 2.3爬虫算法

1. 给定初始url\_start，初始化待爬取url队列Todo，初始化已爬取url列表Visited，初始化已爬取url的MD5码列表Visited\_MD5
2. 将url\_start加入Todo
3. 如果Todo不为空

Todo进行出队操作，出队的url记作url\_todo

计算url\_to的MD5码，判断其是否在Visited\_MD5里

如果是：转入3

如果否：对url\_todo进行爬取，将应网页中所有的url加入到Todo，将url\_to加入Visited,并将url\_todo的MD5码加入Visited\_MD5

4、如果Todo为空，爬虫结束。

# 三、参考文献

[1]成功,李小正,赵全军.一种网络爬虫系统中URL去重方法的研究[J].中国新技术新产品,2014(12):23.

[2]王桦. 基于广度优先的主题爬虫的设计与实现[D].复旦大学,2011.

[3]吴小惠.分布式网络爬虫URL去重策略的改进[J].平顶山学院学报,2009,24(05):116-119.