**爬虫搜索策略-防止环路的出现:**

现在看看图论的遍历算法和搜索引擎的关系。互联网虽然很复杂，但是说穿了其实就是一张大图而已一可以把每一个网页当作一个节点，把那些超链接(Hyperlinks )当作连接网页的弧。网页中那些蓝色、带有下划线的文字背后其实藏着对应的网址，当你点击的时候，浏览器通过这些隐含的网址跳转到相应的网页。这些隐含在文字背后的网址称为”超链接”。有了超链接，我们可以从任何一个网页出发，用图的遍历算法，自动地访问到每一个网页并把它们存起来。完成这个功能的程序叫做网络爬虫( Web Crawlers ).或者在一些文献中称为“机器人”( Robot)。世界上第一个网络爬虫是由麻省理工学院的学生马休·格雷( Matthew Gray)在1993年写成的。他给自己的程序起了个名字叫“互联网漫游者，(WWW Wanderer)。以后的网络爬虫越写越复杂。但原理是一样的。

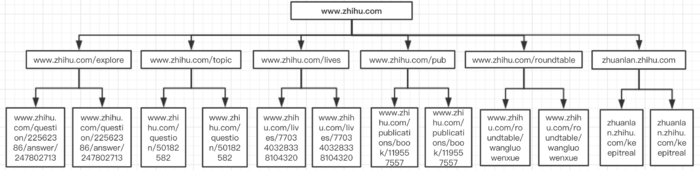
我们来看看网络爬虫如何下载整个互联网。假定从一家门户网站的首页出发。先下载这个网页，然后通过分析这个网页，可以找到页面里的所有超链接。也就等于知道了这家门户网站首页所直接链接的全部网页，诸如雅虎邮件、雅虎财经、雅虎新闻等。接下来访向、下载并分析这家门户网站的邮件等网页，又能找到其他相连的网页。让计算机不停地做下去，就能下载整个的互联网。当然，也要记载哪个网页下载过了，以免重复。在网络爬虫中，使用一个称为“哈希表”( Hash Table )的列表而不是一个记事本记录网页是否下载过的信息。

现在的互联网非常庞大，不可能通过一台或几台计算机服务器就能完成下载任务。比如Google在2010。年时整个的索引大小大约有S 000亿个网页，即使更新最频繁的基础索引也有100亿个网页，假如下载一个网页需要一秒钟，下载这100亿个网页则需要317年，如果下载5 000亿个网页则需要16 000年左右，是我们人类有文字记载历史的三倍时间。因此，一个商业的网络爬虫需要有成千上万台服务器，并且通过高速网络连接起来。如何建立起这样复杂的网络系统，如何协调这些服务器的任务，就是网络设计和程序设计的艺术了。

**2.4.1网站的树结构**

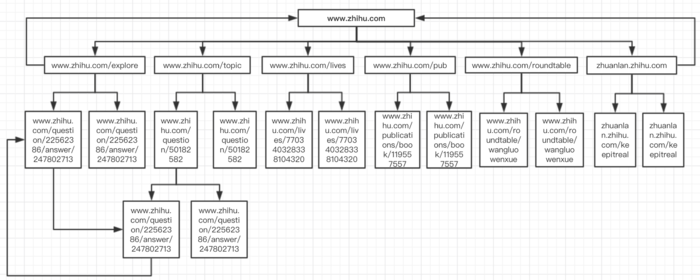
1、一个网站的URL结构图

以知乎为例，知乎目前有发现、话题、Live、书店、圆桌、专栏主要的6个tab页。每个网站的url都是有一定的层次，如下图：发现explore、话题topic、Live lives、书店pub、圆桌roundtable、专栏zhuanlan都是在主域名zhihu的下一级，而具体的Live在…/67006058/answer内容又在话题之下zhihu/question/67006058/answer/250037350，网站的所有内容都一层一层的类似一个树形结构。



2、网站URL链接的结构图

当然，如果我们要做爬取整个网站的url时，我们必须要知道每个网站的url链接一般情况下都是存在环路的，也就是在下一级页面存在上一级页面的url链接，这样形成一个环路。当遇到这个情况时我们需要做url去重，一般的处理方式是把已经爬过的url放到一个list，每次爬取url的时候都去这个list查看下是否已经爬过，爬过的就跳过。这块url去重我下次再详细介绍。



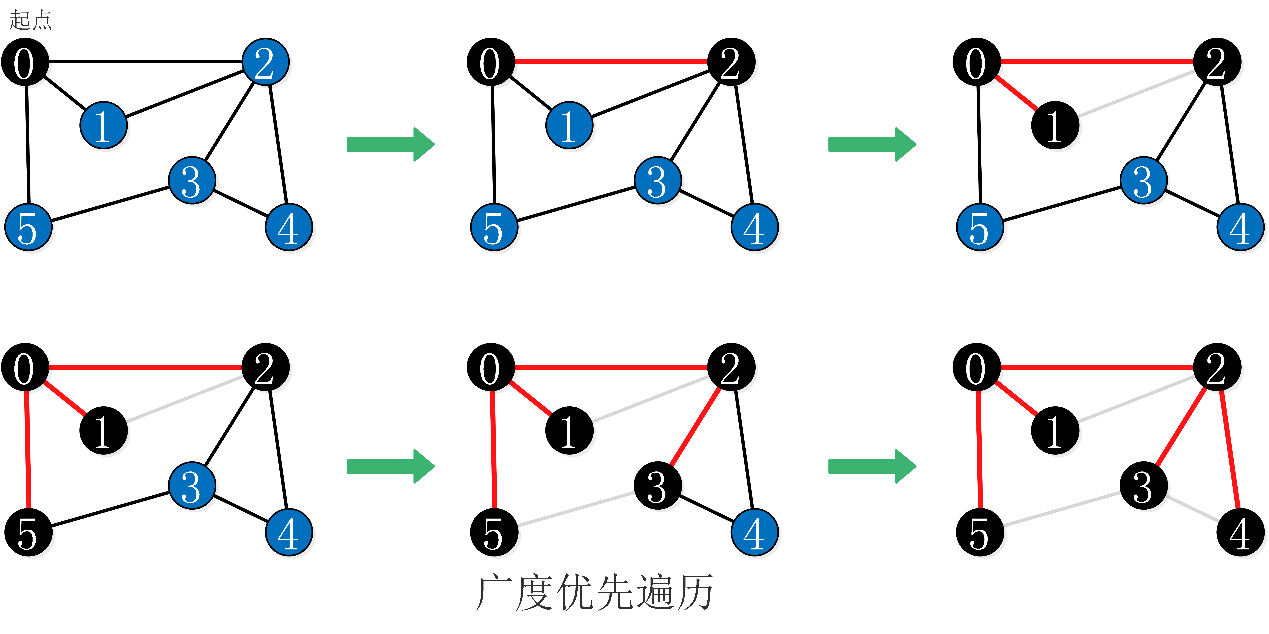
**2.4.2网络爬虫图的搜索策略**：

常见的爬虫爬取策略有以下两种：

1.广度优先遍历策略

广度优先遍历策略，可理解为一种基于网络层次分析的遍历策略，是将网络划分成若干层次，其中，种子站点是处于结构的最顶层。在遍历时，需要处理Web页之间的抓取优先级的问题，一般规定层次结构中层次越高，优先级也就越高;同一层次中，从左到右优先级依次降低。因此，如果只用一个网络爬虫进行遍历操作，那么就按不同层次之间先高后低的次序，以及同层次之间先左后右的次序进行。当然，也可以将多个网络爬虫设计成分布式的结构，它们分别负责其中一层或几层Web站点的遍历，这样不但技术实现的难度不高，而且可以较好地避免重复遍

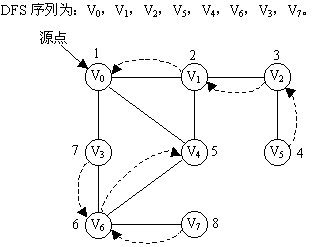
BFS遍历如下图:



2.深度优先策略

深度优先遍历类似于树的前序遍历。它将遍历的侧重点放在网络爬虫爬行的深度上，往往为了抓取更多的Web网页，允许网络爬虫访问站点中某一超链接相关的所有链接。从互联网的结构看，Web页之间通过数量不等的超链接相互连接，形成一个彼此关联、庞大复杂的有向图。所以，按照深度优先的原则，网络爬虫通常需要多次爬行一些Web页，即出现重复爬行现象。如何规避此类问题也就成为修正深度优先策略的重中之重，在实践中，一般会建立一个爬行路径优选算法来简化网络爬虫的行走路线，同时需要依照具体情况确保一个合适的遍历深度。对比以上两种遍历策略errew，广度优先策略更适合大规模Web信息的搜集，可充分发挥网络爬虫程序算法的功效，适合水平型搜索引擎的信息抓取;而宽度优先策略一般适用于垂直搜索引擎的信息搜集。但在实际应用中，二者可相互嵌套。

DFS遍历如下图：



分析：

深度优先搜素算法：不全部保留结点，占用空间少；有回溯操作(即有入栈、出栈操作)，运行速度慢。

广度优先搜索算法：保留全部结点，占用空间大； 无回溯操作(即无入栈、出栈操作)，运行速度快。

通常深度优先搜索法不全部保留结点，扩展完的结点从数据库中弹出删去，这样，一般在数据库中存储的结点数就是深度值，因此它占用空间较少。

所以，当搜索树的结点较多，用其它方法易产生内存溢出时，深度优先搜索不失为一种有效的求解方法。

广度优先搜索算法，一般需存储产生的所有结点，占用的存储空间要比深度优先搜索大得多，因此，程序设计中，必须考虑溢出和节省内存空间的问题。

但广度优先搜索法一般无回溯操作，即入栈和出栈的操作，所以运行速度比深度优先搜索要快些。

综上所述: 我采取了广度优先搜索的爬行，避免深度优先陷入某个站点的环路中，无法访问其他站点。