宽度优先搜索
Breadth First Search

课程版本 v4.0.2 主讲 令狐冲

扫描二维码关注微信/微博
获取最新面试题及权威解答

微信: ninechapter
微博: http://www.weibo.com/ninechapter
知乎: http://zhuanlan.zhihu.com/jiuzhang
官网: http://www.jiuzhang.com

大纲 Outline

- 二叉树上的宽搜 BFS in Binary Tree
- 图上的宽搜 BFS in Graph
  - 拓扑排序 Topological Sorting
- 棋盘上的宽搜 BFS

什么时候应该使用BFS？

图的遍历 Traversal in Graph

- 层级遍历 Level Order Traversal
- 由点及面 Connected Component
- 拓扑排序 Topological Sorting

最短路径 Shortest Path in Simple Graph

- 仅限简单图求最短路径
- 即，图中每条边长度都是1，且没有方向

二叉树上的宽度优先搜索

BFS in Binary Tree

令狐师兄带你写代码之
Binary Tree Level Order Traversal

http://www.lintcode.com/problem/binary-tree-level-order-traversal/

http://www.jiuzhang.com/solutions/binary-tree-level-order-traversal/

图的遍历（层级遍历）

注：树是图的一种特殊形态，树属于图

使用队列作为主要的数据结构 Queue

思考：用栈（Stack）是否可行？为什么行 or 为什么不行？

是否需要实现分层？

需要分层的算法比不需要分层的算法多一个循环

size=queue.size()

如果直接 for (int i = 0; i < queue.size(); i++) 会怎么样？

Binary Tree Serialization (M+Y)

问:什么是序列化？

什么是序列化？

将“**内存**”中结构化的数据变成“**字符串**”的过程

序列化：object to string

反序列化：string to object

什么时候需要序列化？

1. 将内存中的数据持久化存储时

内存中重要的数据不能只是呆在内存里，这样断电就没有了，所需需要用一种方式写入硬盘，在需要的时候，能否再从硬盘中读出来在内存中重新创建

2. 网络传输时

机器与机器之间交换数据的时候，不可能互相去读对方的内存。只能讲数据变成字符流数据(字符串)后通过网络传输过去。接受的一方再将字符串解析后到内存中。

常用的一些序列化手段:

- XML
- Json
- Thrift (by Facebook)
- ProtoBuf (by Google)

序列化算法

一些序列化的例子：

- 比如一个数组，里面都是整数，我们可以简单的序列化为”[1,2,3]”
- 一个整数链表，我们可以序列化为，”1->2->3”
- 一个哈希表(HashMap)，我们可以序列化为，”{\”key\”: \”value\”}”

序列化算法设计时需要考虑的因素：

- **压缩率**。对于网络传输和磁盘存储而言，当然希望更节省。
  - 如 Thrift, ProtoBuf 都是为了更快的传输数据和节省存储空间而设计的。
- **可读性**。我们希望开发人员，能够通过序列化后的数据直接看懂原始数据是什么。
  - 如 Json，LintCode 的输入数据

二叉树如何序列化？

你可以使用任何你想要用的方法进行序列化，只要保证能够解析回来即可。

LintCode 采用的是 BFS 的方式对二叉树数据进行序列化，这样的好处是，你可以更为容易的自己画出整棵二叉树。

算法描述：

http://www.lintcode.com/en/help/binary-tree-representation/

题目及解答：

http://www.lintcode.com/en/problem/binary-tree-serialization/

http://www.jiuzhang.com/solutions/binary-tree-serialization/

相关问题 Related Problems

Binary Tree Level Order Traversal II

http://www.lintcode.com/en/problem/binary-tree-level-order-traversal-ii/

http://www.jiuzhang.com/solutions/binary-tree-level-order-traversal-ii/

Binary Tree Zigzag Order Traversal

http://www.lintcode.com/en/problem/binary-tree-zigzag-level-order-traversal/

http://www.jiuzhang.com/solutions/binary-tree-zigzag-level-order-traversal/

Convert Binary Tree to Linked Lists by Depth

http://www.lintcode.com/en/problem/convert-binary-tree-to-linked-lists-by-depth/

http://www.jiuzhang.com/solutions/convert-binary-tree-to-linked-lists-by-depth/

图上的宽度优先搜索

BFS in Graph

问:和树上有什么区别？

HashMap

图中存在环

存在环意味着，同一个节点可能重复进入队列

Graph Valid Tree

http://www.lintcode.com/problem/graph-valid-tree/

http://www.jiuzhang.com/solutions/graph-valid-tree/

图的遍历（由点及面）

问：如何用基本数据结构表示一个图？

令狐师兄带你写代码之
Clone Graph (F)

http://www.lintcode.com/problem/clone-graph/

http://www.jiuzhang.com/solutions/clone-graph/

图的遍历（由点及面）

Search Graph Nodes (A)

http://www.lintcode.com/problem/search-graph-nodes/

http://www.jiuzhang.com/solutions/search-graph-nodes/

图的遍历（由点及面）

问：为什么不需要做分层遍历？

follow up: 如何找***所有***最近的value=target的点？

Topological Sorting

http://www.lintcode.com/problem/topological-sorting/

http://www.jiuzhang.com/solutions/topological-sorting/

几乎每个公司都有一道拓扑排序的面试题！

问：可以使用 DFS 来做么？

独孤九剑——破枪式

能够用 BFS 解决的问题，一定**不要**用 DFS 去做！

Course Schedule I && II (G+A+F+Z)

http://www.lintcode.com/en/problem/course-schedule/

http://www.lintcode.com/problem/course-schedule-ii/

裸拓扑排序

Sequence Reconstruction (G+A)

http://www.lintcode.com/problem/sequence-reconstruction/

判断是否只存在一个拓扑排序的序列

只需要保证队列中一直最多只有1个元素即可

矩阵中的宽度优先搜索

BFS in Matrix

矩阵 vs 图

图 Graph

N个点，M条边

M最大是 O(N^2) 的级别

图上BFS时间复杂度 = O(M)

所以最坏情况可能是 O(N^2)

矩阵 Matrix

N行M列

N*M个点，N*M*2 条边（每个点上下左右4条边，每条边被2个点共享）。

矩阵中BFS时间复杂度 = O(N * M)

Number of Islands

http://www.lintcode.com/problem/number-of-islands/

http://www.jiuzhang.com/solutions/number-of-islands/

图的遍历（由点及面）

坐标变换数组

```
int[] deltaX = {1,0,0,-1};
int[] deltaY = {0,1,-1,0};
```

问：写出八个方向的坐标变换数组？

Zombie in Matrix

http://www.lintcode.com/problem/zombie-in-matrix/

http://www.jiuzhang.com/solutions/zombie-in-matrix/

图的遍历（层级遍历）

Knight Shortest Path

http://www.lintcode.com/problem/knight-shortest-path/

http://www.jiuzhang.com/solutions/knight-shortest-path/

简单图最短路径

follow up: speed up?

Build Post Office II

http://www.lintcode.com/problem/build-post-office-ii/

http://www.jiuzhang.com/solutions/build-post-office-ii/

简单图最短路径

哪种方法更好？

方法1：从空格出发

- 循环枚举所有的office修建位置的可能性（空格）
- 计算从这个位置出发到达所有房子的距离之和
- 在所有方案中找到最小的距离和

方法2：从房子出发

- 循环枚举所有的房子的位置
- 从房子出发，计算每个空格到达房子的距离
- 累加某个空格到达其他所有房子的距离之和
- 在所有空格中，找到最小距离和

图的遍历(由点及面)

- 无向图联通块
- http://www.lintcode.com/problem/connected-component-in-undirected-graph/
- 覆盖黑点的最小矩阵(BFS无法AC但是可以作为BFS的练习题)
- http://www.lintcode.com/problem/smallest-rectangle-enclosing-black-pixels/

简单图最短路径

- 单词阶梯
- http://www.lintcode.com/problem/word-ladder/

总结 Conclusion

- 能用 BFS 的一定不要用 DFS（除非面试官特别要求）
- BFS 的两个使用条件
  - 图的遍历（由点及面，层级遍历）
  - 简单图最短路径
- 是否需要层级遍历
  - size = queue.size()
- 拓扑排序必须掌握！
- 坐标变换数组
  - deltaX, deltaY
  - inBound