数据结构
Data Structure

课程版本 v3.3　　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

- Linear Data Structure
  - Queue
  - Stack
  - Hash
- Tree Data Structure
  - Heap / Priority Queue
  - TreeMap

What is Data Structure?

可以认为是一个集合，并且提供集合上的若干操作

队列 Queue

支持操作：O(1) Push / O(1) Pop / O(1) Top

BFS的主要数据结构

多做做BFS的题就可以了

栈 Stack

支持操作：O(1) Push / O(1) Pop / O(1) Top

非递归实现DFS的主要数据结构

独孤九剑 —— 破箭式

BFS 的主要数据结构是 Queue

DFS 的主要数据结构是 Stack

千万不要搞反了！

很体现基础知识的扎实度！

Min Stack

http://www.lintcode.com/problem/min-stack/

http://www.jiuzhang.com/solutions/min-stack/

Largest Rectangle in Histogram

http://www.lintcode.com/problem/largest-rectangle-in-histogram/

http://www.jiuzhang.com/solutions/largest-rectangle-in-histogram/

为什么这个题不能使用动态规划？

Related Questions

- Maximal Rectangle (histogram近似题)
- http://www.lintcode.com/problem/maximal-rectangle/
- http://www.jiuzhang.com/solutions/maximal-rectangle/
- Max Tree (histogram近似题)
- http://www.lintcode.com/problem/max-tree/
- http://www.jiuzhang.com/solutions/max-tree/
- Implement Stack by Two Queues
- http://www.lintcode.com/problem/implement-stack-by-two-queues/
- http://www.jiuzhang.com/solutions/implement-stack-by-two-queues/
- Implement Queue by Two Stacks
- http://www.lintcode.com/problem/implement-queue-by-two-stacks/
- http://www.jiuzhang.com/solutions/implement-queue-by-two-stacks/

Take a break

5 minutes

哈希表 Hash

支持操作：O(1) Insert / O(1) Find / O(1) Delete

Hash Table / Hash Map / Hash Set 的区别是什么？

Hash Function

使命：对于任意的key

得到一个*固定*且*无规律*的介于*0~capacity-1*的整数

Hash Function

- 一些著名的Hash算法
  - MD5
  - SHA-1
  - SHA-2

```
int hashfunc(String key) {
    return md5(key) % hash_table_size;
}
```

- 以 String 为例子

```
int hashfunc(String key) {
    int sum = 0;
    for (int i = 0; i < key.length(); i++) {
        sum = sum * 31 + (int)(key.charAt(i));
        sum = sum % HASH_TABLE_SIZE;
    }
    return sum;
}
```

Open Hashing vs Closed Hashing

再好的 hash 函数也会存在冲突(Collision)

https://www.cs.usfca.edu/~galles/visualization/ClosedHash.html

https://www.cs.usfca.edu/~galles/visualization/OpenHash.html

Rehashing

当hash不够大时怎么办？

http://www.lintcode.com/problem/rehashing/

http://www.jiuzhang.com/solutions/rehashing/

LRU Cache

http://www.lintcode.com/problem/lru-cache/

http://www.jiuzhang.com/solutions/lru-cache/

Example: [2 1 3 2 5 3 6 7]

LRU Cache

- LinkedHashMap = DoublyLinkedList + HashMap

- HashMap<key, DoublyListNode> DoublyListNode {
- prev, next, key, value;
- }

- Newest node append to tail.
- Eldest node remove from head.

Related Questions

- http://www.lintcode.com/problem/subarray-sum/
- http://www.lintcode.com/problem/copy-list-with-random-pointer/
- http://www.lintcode.com/problem/anagrams/
- http://www.lintcode.com/problem/longest-consecutive-sequence/

Heap

支持操作：O(log N) Add / O(log N) Remove / O(1) Min or Max

Max Heap vs Min Heap

PriorityQueue vs Heap

Heap 的基本原理和具体实现

我们放到了九章算法强化班中

基本操作——Heapify

http://www.lintcode.com/problem/heapify/

http://www.jiuzhang.com/solutions/heapify/

https://www.cs.princeton.edu/~wayne/kleinberg-tardos/pdf/DemoHeapify.pdf

Ugly Number

http://www.lintcode.com/problem/ugly-number-ii/

http://www.jiuzhang.com/solutions/ugly-number-ii/

Top k Largest Number II

http://www.lintcode.com/problem/top-k-largest-numbers-ii/

http://www.jiuzhang.com/solutions/top-k-largest-number-ii/

Related Questions

- http://www.lintcode.com/problem/merge-k-sorted-lists/
- http://www.lintcode.com/problem/merge-k-sorted-arrays/
- http://www.lintcode.com/problem/data-stream-median/
- http://www.lintcode.com/problem/top-k-largest-numbers/
- http://www.lintcode.com/problem/kth-smallest-number-in-sorted-matrix/

TreeMap

又想知道最小值，又想支持修改和删除

https://docs.oracle.com/javase/7/docs/api/java/util/TreeMap.html

相关习题

- http://www.lintcode.com/problem/building-outline/
- http://www.lintcode.com/problem/top-k-frequent-words/