Data Structure

现在是课程答疑时间

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
官网: www.jiuzhang.com

Outline

- Linear Data Structure
  - Queue
  - Stack
  - Hash
- Tree Data Structure
  - Heap

What is Data Structure

Data Structure is a way to organize data.

It provides some methods to handle data stream,

e.g. insert, delete, etc.

Queue

Operations

- O(1) Push
- O(1) Pop
- O(1) Top

Core data structure for BFS!

Stack

Operations:

- O(1) Push
- O(1) Pop
- O(1) Top

Min Stack

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

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

Implement a Queue by Two Stacks

http://www.lintcode.com/problem/implement-queue-by-two-stacks/

http://www.jiuzhang.com/solutions/implement-queue-by-two-stacks/

Largest Rectangle in Histogram

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

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

Max Tree

http://www.lintcode.com/problem/max-tree/

http://www.jiuzhang.com/solutions/max-tree/

Take a break

九章算法

5 minutes break

Hash

Operations

- O(1) Insert
- O(1) Delete
- O(1) Find

Hash Function

Collision

- Open Hashing (LinkedList)
- Closed Hashing (ArrayList)

Hash Function

Typical: From string to int.

```
int hashfunc(String key) {
   // do something to key
   // return a deterministic integer number
   return md5(key) % hash_table_size;
}
```

Hash Function - Magic Number 33


```
int hashfunc(String key) {
    int sum = 0;
    for (int i = 0; i < key.length(); i++) {
        sum = sum * 33 + (int)(key.charAt(i));
        sum = sum % HASH_TABLE_SIZE;
    }
    return sum;
}
```

Open Hashing vs Closed Hashing

Rehashing

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

Java

- HashTable
- HashSet
- HashMap

Which on is Thread Safe?

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

Operations

- O(log N) Add
- O(log N) Remove
- O(1) Min/Max

Median Number

http://www.lintcode.com/problem/data-stream-median/

http://www.jiuzhang.com/solutions/median-in-data-stream/

Related Questions

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

http://www.lintcode.com/problem/merge-k-sorted-lists/

http://www.lintcode.com/problem/merge-k-sorted-arrays/

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