

cs271 Exam 2 Review Guide

Topics

- Stacks (array & linked list)
 - Queues (array & linked list)
 - Priority Queues
 - Recursion
 - Quick Sort
 - Merge Sort
 - Towers of Hanoi
 - P vs NP
 - Tree Terminology and Representation
 - Binary Search Trees
 - Parent Pointers in Trees
 - Tree Rotations
 - Red Black Trees
 - insertions
 - AA Trees
 - AVL Trees
 - Splay Trees
- (reminder: jhave available for trees and sorts)

Specifically not on Exam 2

- computing Big O of complicated recursive functions
- Red Black Tree Removals
- AA Tree operations (insert, remove, etc.)
- AVL Tree operations (insert, remove, etc.)
- Splay Trees (insert, remove, etc.)

Sample Exam Questions:

- 1.) Draw step by step the merge sort of the following array ...
[5 7 3 2 4 9 8 1]
- 2.) Draw step by step the quick sort of the following array, assume we pick the best pivot
[6 5 3 4 1]
- 3.) Explain 2 disadvantages of using Recursion instead of iteration.
- 4.) Explain the different between a stack and a queue.
- 5.) True or False : Stack is LIFO.
- 6.) Explain why it would be such a huge breakthrough if somebody discovered a Polynomial time algorithm for an NP problem.
- 7.) Using this tree in parenthetical notation, draw the tree in general tree format.
A (B (C D E) F (G H) I (J))
- 8.) What is the height of the tree in #7?
- 9.) What is the degree of Node F in tree # 7? What is the indegree? Outdegree? Level?
- 10.)List the Leaf nodes from #7.
- 11.)Which traversal on a Binary Search Tree gives us a sorted list?
 - a.) Pre Order Traversal
 - b.) In Order Traversal
 - c.) Post Order Traversal
- 12.)What is the Pre-Order traversal on #7?
- 13.)What is the Post-Order traversal on #7?
- 14.)What is the In-Order traversal of this binary tree?
A (B (D) E (F (G) H (I)))

15.) How would you write a C++ function that prints a binary search tree in indented notation?
IndentationFormatHelper(this->root, 0);

```
void IndentationFormatHelper(node * cur, int indent)
{

}

}
```

16.) Complete this recursive Insert function for a Binary Search Tree
InsertHelper(this->froot, data);

```
void InsertHelper(node * cur, type data)
{

}

}
```

17.) Fill in the rest of the 5 rules for Red Black Trees

1. Every node is either Red or Black
- 2.
3. All Leaves are Black
- 4.
- 5.

18.) Explain the benefit of a Red Black Tree over a Binary Search Tree

19.) Perform a “right rotation” on Node B in this tree and show the resulting tree

A (B (C (D E)) G)

20.) Perform a “left rotation” on Node B in this tree and show the resulting tree

A (B (E (F K)) G)

21.) Explain what's different in a splay tree than other self-balancing trees?