|  |  |
| --- | --- |
| Name |  |
| KUID |  |

# EECS 268 2020 Fall: Final Exam

Possible points: 100pts

Time Limit: 2.5 hours

Submit via email to your Lab TA

Place an X in the box that applies to you:

|  |  |
| --- | --- |
| I am in the 12pm section (officially) |  |
| I am in the 1pm section (officially) |  |

# 

# Rules

# 

**You will receive a zero for this exam unless you read the following and answer below**

Hi students,

I know this has been a tough, stressful, completely unprecedented semester. I really do appreciate all the effort you've put into this course.

Unfortunately, before we get started I need to remind everyone of the following rules:

* You must take this exam by yourself, no communicating with anyone else during the exam
* DO NOT use unauthorized aid such as chegg, google searches, exams from past semesters, coding forums, etc.
* DO NOT use a compiler

You 100% are allowed to look at:

* Any notes, code, or videos from the class wiki
* Your notes and code

In short, I need you to agree that you will take this exam by yourself and in an honest way. I need you to think of someone in your life, other than me, that would be sorely disappointed if they knew you cheated and write their name below.

*I agree to follow the rules for this exam because I want to do right by the following person…*

|  |  |
| --- | --- |
| Answer |  |

# [10pts] Maze Walking

Given the following maze and starting coordinates, what is the exhaustive order (no unmarking after deadends) that we will traverse the maze.

Rules:

P - passage

W - Walls

E - Exit (as soon as exit is found, stop)

Move order check:

Up, Right, Down, Left

Starting position: 4, 2 (bottom row, third column, I've bolded it)

Maze

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| E | P | P | P | P | P |
| W | W | P | W | W | P |
| P | W | P | W | P | P |
| P | W | P | W | W | P |
| P | P | **P** | P | W | P |

Edit this table to show the order of traversal (remember, don't just show the final path, but the order we'd visit all possible passageways until an exit is found. I've done the first 3 steps for you.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 3 | 0 | 0 | 0 |
| 0 | 0 | 2 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 |

# 

# [10pts] Code writing: Heaps

Assume you are in a MinHeap.cpp. This class is an array-based implementation of a Min Heap and has the following member variables:

* int m\_arraySize
* int m\_heapSize
* T\* m\_arr

Define a private, recursive downheap method that would appropriately assist the given public method:

|  |
| --- |
| template <typename T>  void MinHeap<T>::remove()  {  //swap the lowest right-most  //to the root then downheap it  m\_heapSize--;  T temp = m\_arr[0];  m\_arr[0] = m\_arr[m\_heapSize];  m\_arr[m\_heapSize] = temp;  downheap(0);  } |

Your code will be on the following page

|  |
| --- |
| template <typename T>  void MinHeap<T>::downheap(int index)  {  //your code below |

# [10pts] Code Writing: Recursion

There is a nest of bugs that lives underneath my floor. They are multiplying at an alarming rate. On the first day, there were only 2 bugs. On the second day there were 7. On the third day, there were 91. Every day since, the number of bugs underneath my floor is equal to the previous three days combined. Write a recursive function that will take in a day and return the number of bugs that I will find under my floor that day (e.g. bugs(1) returns 2, bugs(4) returns 100).

|  |
| --- |
| int bugs(int day) {  //Your definition here |

# [15pts] Code writing: Binary Trees

Assume you are in a Binary Tree implementation. The only private member variable is a BNode<T>\* m\_root. You must define a method, countRightChildren, that returns the number of nodes that are a right child. You will define a public facing method that takes zero parameters and a private recursive method that takes a BNode<T>\* representing a subtree. Assume the BNode class has getLeft() and getRight() methods for retrieving left and right pointers.

## [5pts] Binary Trees: part 1

|  |
| --- |
| template <typename T>  int BinaryTree<T>::countRightChildren() const  {  //your definition below |

## [10pts] Binary Trees: part 2

|  |
| --- |
| template <typename T>  int BinaryTree<T>::countRightChildren(BNode<T>\* subtree) const  {  //your definition below |

# [12pts] Binary Trees

[3pts each] Confirm all that apply to the following graph (yes/no):

1. Is it a valid tree?

|  |
| --- |
|  |

1. Is it a valid binary tree?

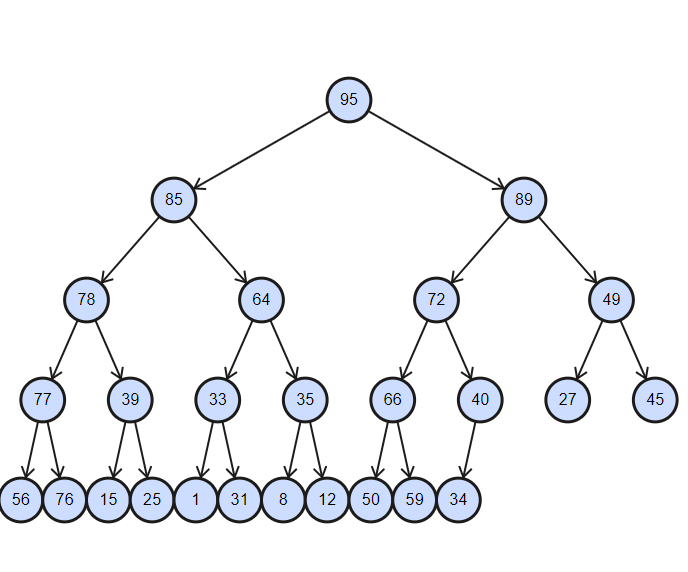
|  |
| --- |
|  |

1. Is it a valid Binary Search Tree?

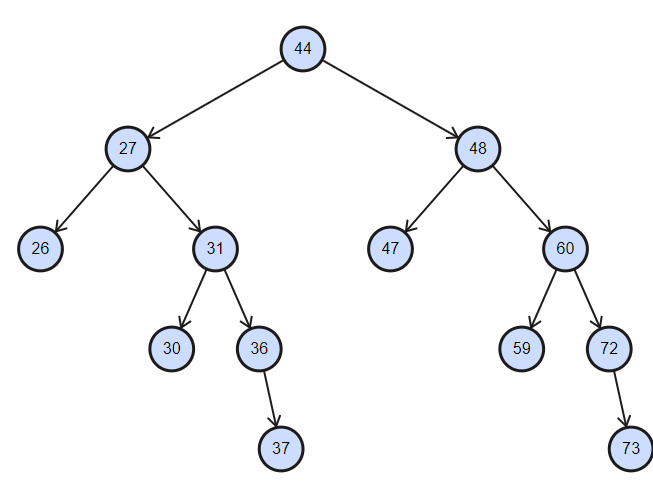
|  |
| --- |
|  |

1. Is it a valid Min Heap?

|  |
| --- |
|  |



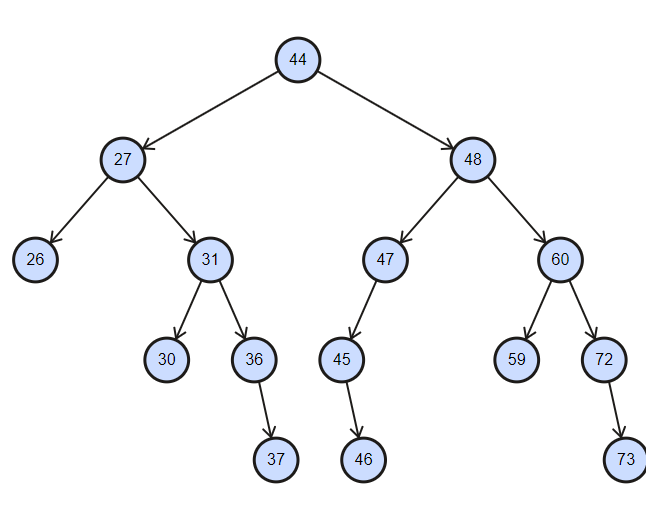
# [15pts] Binary Search Trees: Traversal



Write the order in which the nodes would be visited in Pre, In, and Post order traversal. Each traversal is **all or nothing**, so double check your work.

|  |  |
| --- | --- |
| Traversal [5pts each] | Values |
| Pre |  |
| In |  |
| Post |  |

# [6pts] Binary Trees: Removal

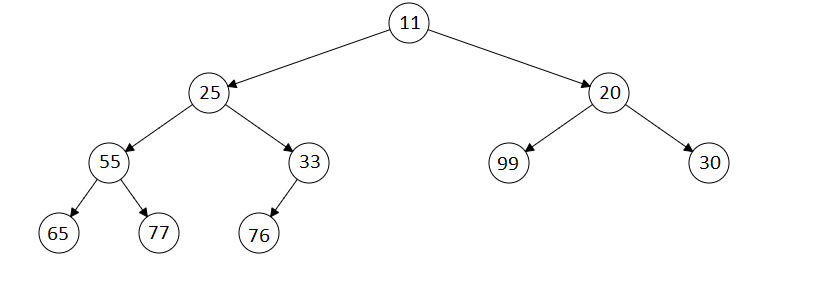


Below, write the pre-order traversal of the resulting Binary Search Tree if 44 is removed. The replacement candidate (if needed) should be the **minimum value of the right subtree**. (This is **all or nothing**)

|  |  |
| --- | --- |
| Pre order: |  |

# 

# [6pts] Heaps



1. [2pts] Given this min heap, fill in the value assuming an array based implementation of the heap. The top row represents the indices of the array. You write the values in the second row. Leave any undefined indices blank.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** |
|  |  |  |  |  |  |  |  |  |  |

1. [4pts] Now, fill in the value assuming an array based implementation of the heap after 3 removes occur. The top row represents the indices of the array. You write the values in the second row. Leave any undefined indices blank.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** |
|  |  |  |  |  |  |  |  |  |  |

# [16pts] Conceptual: Short Answer

[2pts] Where is the smallest value in a min heap?

|  |
| --- |
|  |

[2pts] Where is the largest value in a min heap?

|  |
| --- |
|  |

[2pts] If bubble sort makes a pass through an array without making any swaps, what, if anything, does that tell us about the array?

|  |
| --- |
|  |

[2pts] What does the helper function merge(), which is used by merge sort, assume/require about the two arrays being passed in?

|  |
| --- |
|  |

[6pts] Assume you have a BST with at least two nodes. Where could the second smallest be? List all possibilities.

|  |
| --- |
|  |

[2pts] Which traversal order is most appropriate for a Tree's destructor (Pre, in, or post order)?

|  |
| --- |
|  |

This should be page 15.

If it's not, please make sure you didn't alter the format.

The final is over...or is it?

It is.

Good work.

Email it to your TA.