# EECS 268 2020 Spring Final Exam

Possible Points: 114pts

Due Date: Wednesday May 13th 11:59:59am (noon)

Submit via email to your Lab TA

Place an X in the box that applies to you:

|  |  |
| --- | --- |
| I am in the MWF section (officially) |  |
| I am in the TR section (officially) |  |

# Next, read and sign the following page.

# Rules (Read These!)

* DO NOT alter the formatting of the test. Any changes to formatting could result in grading errors
* Only mark your answers within designated answer boxes
* Read and sign below

Hi EECS 268 students. I know you just got this take home exam off the internet, but can you promise me, or more importantly, promise yourself that you'll take this exam without any unauthorized aid?

* Unauthorized aid: google searches, the materials or help of other students, past exams, help from the undergrad staff or GTAs, any compilers (e.g. don't just put the code tracing problems in a compiler and run them)

But since we're in such a crazy world and our class didn't go as planned, I am fine with you using the following authorized aid:

* Authorized resources: your notes, your labs (the code, not the compiler), the online lectures I made for you, your amazing brain

I'd like this exam to still be an assessment of your skill and understanding.

If you agree to this, then please type your name in the box below. If you can't agree to it, then I'm afraid I won't grade your exam. Afterall, I'm not here to test how well you can google things. That's what your job will do.

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

# Provided Code: LinkedList.h

For questions that ask you to use a LinkedList, below are the public methods you may call.

REMINDER: Indexing starts at 1 in our Linked List

|  |
| --- |
| #include "Node.h"  #include "ListInterface.h"  #include <stdexcept> //For runtime\_error  template <typename T>  class LinkedList : public ListInterface<T> {  private:  Node<T>\* m\_front;  int m\_length;    public:  LinkedList();  LinkedList(const LinkedList& original);  ~LinkedList();  void operator=(const LinkedList& original);  bool isEmpty() const;  int getLength() const;  void insert(int position, T entry); //can throw runtime\_error  void remove(int position); //can throw runtime\_error  void clear();   T getEntry(int position) const; //can throw runtime\_error  void replace(int position, T newEntry); //can throw runtime\_error };  #include "LinkedList.cpp" #endif |

# 

# [10pts] Code Writing: Using Lists

Assume you are in main.cpp. Define the free-floating function (not scoped to any class) called *replaceAll.* This function takes a LinkedList by reference, and two values. It then replaces all occurrences of *targetValue* with *replacementValue*.

Example, if the original list contained: 1,3,2,4,3,9,5

then a call to replaceAll looked like: *replaceAll( originalList, 3, 9);*

the list would then contain: 1,9,2,4,9,9,5

You may assume that the type T has a working copy constructor and overloaded assignment operator, and that you can declare a T like this:

T temp;

REMINDER: You may only use the LinkedList methods provided earlier in the exam.

|  |
| --- |
| //inside main.cpp (NOT a member method of LinkedList!)  template <typename T>  void replaceAll(LinkedList<T>& list,  T targetValue,  T replacementValue){  //Your definition below |

# [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\_array

Define a private upheap method that would appropriately assist the given public add method:

|  |
| --- |
| template <typename T>  void MinHeap<T>::add(T entry)  {  if(m\_size <= m\_heapSize)  {  resize(); //increases heap size to be adequate  }  m\_array[m\_heapSize]= entry;  upheap(m\_heapSize);  m\_heapSize++;  } |

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

# [10pts] Code Writing: Recursion

Define a recursive function to calculate the nth Gibbonnacci number. The Gibbonnacci sequence is a lesser-known numerical sequence that is defined recursively:

Assuming n>= 1 there are some set values for n when n is 1, 2, or 3...

Gibb( 1 ) ⇒ 3

Gibb( 2 ) ⇒ 10

Gibb( 3 ) ⇒ 1983

For all n > 3, it is defined recursively…

Gibb( n ) ⇒ Gibb(n-1) + Gibb(n-2) + Gibb(n-3)

Below, write a recursive definition (NO LOOPS ALLOWED) for calculating the nth number in the Gibbonnacci sequence. You may not alter the parameter list.

|  |
| --- |
| int Gibb( int n) {  //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, *countFullParents*, that returns the number of nodes that have exactly two (2) children. You will define a public facing method that takes zero parameters and a private recursive method that takes a BNode<T>\* representing a subtree.

## [5pts] Binary Trees: part 1

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

## [10pts] Binary Trees: part 2

|  |
| --- |
| template <typename T>  int BinaryTree<T>::countFullParents(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?

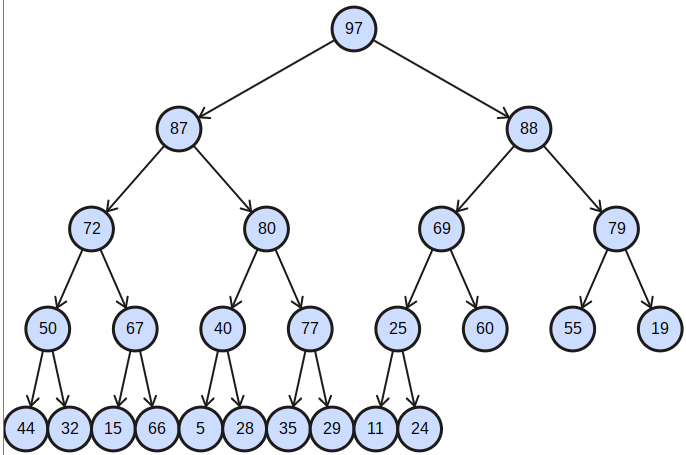
|  |
| --- |
|  |

1. Is it a valid Binary Search Tree?

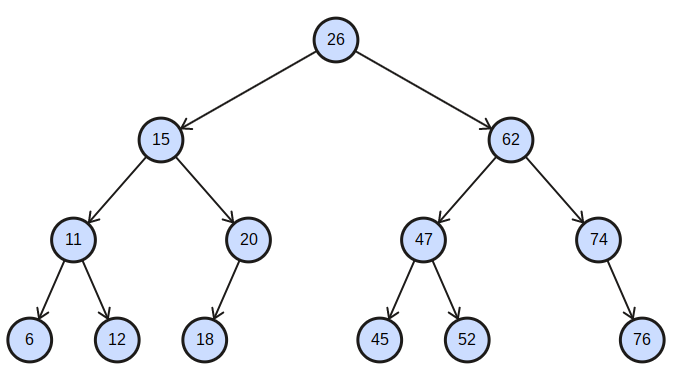
|  |
| --- |
|  |

1. Is it a valid Max Heap?

|  |
| --- |
|  |



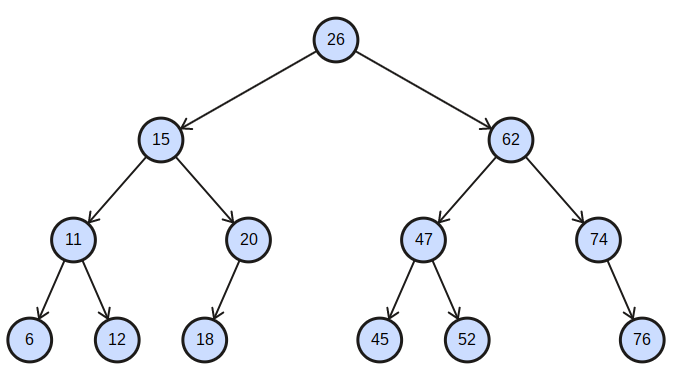
# [15pts] Binary Search Trees: Traversal



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 |  |

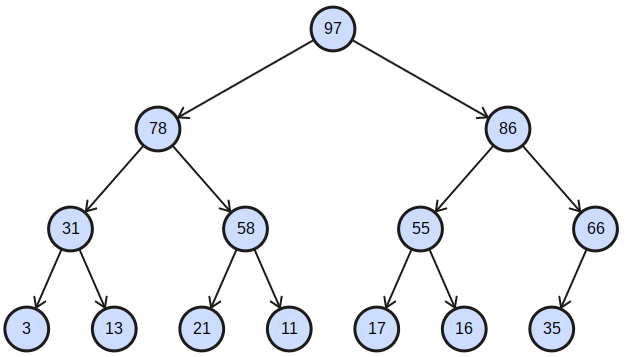
# [6pts] Binary Trees: Removal



Below, draw the resulting Binary Search Tree if 26 is removed. The replacement candidate (if needed) should be the maximum value of the left subtree. (This is **all or nothing**)

|  |
| --- |
|  |

# [6pts] Heaps



[6pts] Given this heap, draw the resulting heap after two removes. You may do scratch work on your own scratch paper, but only put the result after both removes are complete. (This is **all or nothing**)

|  |
| --- |
|  |

# [20pts] Conceptual: Short Answer

[2pts] How does one traverse to the largest (maximum) value in a binary search tree?

|  |
| --- |
|  |

[2pts] How does one traverse to the smallest value in a binary search tree?

|  |
| --- |
|  |

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

|  |
| --- |
|  |

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

|  |
| --- |
|  |

[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 assume/require about the two arrays being passed in?

|  |
| --- |
|  |

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

|  |
| --- |
|  |

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

|  |
| --- |
|  |

# [10pts] Inheritance

Given the relationships between the following classes, answer the following questions. Assume all classes are well defined. The order the information is given is intentional, you'll have to mentally construct the class hierarchy. You DO NOT have to draw the hierarchy.

* A Shark is a Fish
  + All Sharks hunt
* An Owl is a Bird
  + All Owls can hunt
* A Fish is an Animal
  + All Fish can swim
  + Not all Fish hunt
* A Bird is an Animal
  + All Birds can Fly
  + Not all Birds hunt
* All Animals can sleep and eat
  + Not all Animals hunt

1. [3pt] Assume a function playWithAnimal takes an Animal& as a parameter, list all types of objects you can pass to it.

|  |
| --- |
|  |

1. [2pt] What methods can be call directly from playWithAnimal ?

|  |
| --- |
|  |

1. [3pts] If a function, *playWithShark* takes a Shark& as a parameter, list all methods that can be called from it.

|  |
| --- |
|  |

1. [1pt] If a function playWithOwl takes an Owl& as a parameter, can a Bird be passed to it?

|  |
| --- |
|  |

1. [1pt] If a function playWithOwl takes an Owl& as a parameter, can a Shark be passed to it?

|  |
| --- |
|  |

This should be page 16.

If it is not, place make sure you didn't change the size of any of the answer fields.

The exam is now over.

Congratulations!

Email your exam to your TA by 11:59:59am (noon) Wednesday May 13th