Quiz 2 Study Guide

Below is description of work done since the last quiz, including sample problems you can work on. The quiz will have 3 or 4 questions which will fit into the following three categories: (1) Programming, where you will be asked to program in correct C++ syntax with pen and paper; (2) Short Answer, where you will be asked to answer (in paragraph form) a word problem; (3) Math Proof/Calcuation, where you will be asked to provide a short proof or perform a calculation, with work shown.

1 C++ Programming

Inheritance

- What is inheritance, and what is its role in the object hierarchy?
- If `A` is a super class `B` , write (in C++ syntax) the class descriptions to indicate this relationship.
- Describe the security flags `public`, `private`, and `protected` and their meaning with respect to inheritance
- Describe the meaning of the `virtual` flag with respect to inheritance. Provide an example where `virtual` is neccesary to ensure that the *most specific* member function of a class is used.

Abstract/Pure-Virtual Classes

- What does it mean for a class to be pure virtual?
- What is the syntax to indicate a member function of a class is pure virtual?
- Can pure virtual member functions and non-virtual member functions be mixed? Provide a trivial example of this that must be the case for all classes, regardless of being pure-virtual.
- Why are pure virtual classes relevant to data structures? Provide an example from class of where pure virtual classes play a significant role.

2 Algorithm Analysis

Big-O Analysis

- Write the mathematical deffinition of Big-O?
- In general terms, when analyzing a program, what is the process used to determining the Big-O of computation?
- What counts as an operation in Big-O computation?
- Consider a program with two routines, the first routine `foo()` is $O(n)$ and the second routine `bar()` is also $O(n)$. What is the big-O of a program that executes these routines consecutively, e.g., like below:

```
foo()
bar()
```

- What is the Big-O of the below routine, using `bar()` and `foo()` from above?

```
for(i=0 --> n):
    foo()
    bar()
```

- What is the Big-0 of the following small program?

```
for(j=0; j < n ; j++){
   for(i=0;i<n;i*=2){
      cout << i << endl;
   }
}
```

- Order the following functions in terms of there relative magnatude: n, n^2, 1, n^n, $lg(n)$, $n!$, 2^n, $n * lg(n)$, n^4.
- Prove the following is $O(n^2)$: $-3n^2 + 4$

Discreet Proofs

- Describe, in general terms, the proof technique of induction?
- Prove the following using induction:

$$\sum_{i=0}^{n} i = \frac{n(n+1)}{2}$$

- How is analysis using a loop invariant and induction related?

- Show, using a loop invariant and induction, that the following short program verifies that all numbers in the input list are even:

```
bool all_even(int input[], int length){
  for(int i=0; i < length; i++){
    if(i%2){
      return false;
    }
  }
  return true;
}
```

3 Ordered Structures

Lists, ArrayLists and LinkedLists

- Describe the desired properties of a List, as compared to the properties of basic arrays?
- What are the core functions associated with a list? Write out the pure-virtual class definition of a list.
- Describe the implementation of an ArrayList? What is the Big-O of each of the member functions?
- Describe the implementation of a LinkedList? What is the Big-O of each of the member functions?
- When is an ArrayList preferred over a LinkedList, when is a LinkedList preferred over an ArrayList? What properties of each list implementation make it desirable in these situations.
- When expanding an ArrayList, which is more desirable: increase the array by a fixed amount for each expansion, or double the amount of space upon each expansion? Why?
- Using *amortized* analysis, show that doubling the amount of space provide constant average time for array expansion. Similarly, show that fixed amount expansion requires linear average time.

Stacks and Queues

- What is LIFO? What is FIFO? Which operation maps to a queue and which to a stack?
- Provide one example in computer science where a Stack is used. Provide one example in computer science where a Queue is used.
- What are the basic operations of a stack?

- What are the basic operations of a queue?
- Which list implemenation works best for stacks and queues? Why?
- If you were to use an ArrayList to implement a queue, how would you do so to minimize cost? Particularly, what is expensive about using an ArrayList?
- Describe a CircularArrayList and discuss why it avoids the above cost when used in implementing a Queue.

4 Sorting

Polynomial Sorting vs. Logorithmic Sorting

- Describe insertion sort and show that this is a $O(n^2)$ operation.
- Describe merge sort and (by using visuals) show that merge sort is $O(n * lg(n))$.

Pivot/Quick Sorting

- Describe the properties of pivot sorting as compared to insertion sort and merge sort?
- What is a better pivot selection strategy: Always choosing the first item in the sub list as the pivot? Or, choosing a random value as the pivot?
- Explain why Quick Sort is $O(n * lg(n))$?

5 Dynamic Programming

- What is dynamic programming and what kinds of problems does it solve?
- Consider Edit-Distance, which is defined as the number insertions, deletions, and swaps that are required to transform one string into another. For example, the edit distance between "ACG" and "AGC" is two, requiring the swap of "C" to "G" and "G" to "C". And the edit distance between "ACG" and "AGCA", is three, since the same swaps are required, but with an additional insert of "A".

  Explain why dynamic programming is a good solution to this algorithm. Consider how comparing two strings with one additional letter could use information about the comparison of the previous sub-strings.
- Fill in the dynamically constructed matrix for finding the edit distance between "john" and "janet" such that each cell contains the edit distance between the subs-string on the column compared to the sub-string of the row.

	j	a	n	e	t
j					
o					
h					
n					

- Describe the dynamic part of this algorithm. How can each cell be filled in based on information provided by adjacent cells?
- What value in the matrix indicates the edit-distance between the two strings? Explain.
- Argue that finding edit distance of any two strings is $O(n * m)$, where n and m is the length of the strings.