CIS 55
Final Exam, Dec. 16, 2021 75 points total plus a bonus 10 points

True/False (30 pts)
Select true or false for the statements below. Explain your answers if you like to receive partial credit

1) Which of the following is true about searching elements in an unordered array?
    a. With the data is unsorted, search is O(n) because if the element you are looking for is not there, you have to check every element in the array
    b. If you start at the end of the array and traverse to index 0, search improves to O(log n) because you only have to look at half of the array
    c. If you get lucky with checking the first element and find it immediately, then the worst case performance of search improves to O(n^2)

2) Which of the following is true about searching elements in an ordered array?
    a. You cannot use binary search on an ordered array so the performance is O(n)
    b. If there are no holes in the array and the elements are all next to each other, then the performance for search improves to O(1)
    c. If you do a linear search on a sorted array, the performance improves to O(log n)

3) Which of the following is true about the insertBeforeCurrent function for a Circular Linked List (CLL) like you did in programming exercise 1?
    a. If the CLL is empty, you need to create the new node, set it to current, and have its next pointer refer to itself
    b. The worst case performance of the function is O(n)
    c. If you insert a new element with the same data value as the current node, the performance improves to O(log n)

4) Given a stack implemented as a standard linked list (only a head pointer) with O(1) push and O(1) pop, and the following functions are performed in sequence:

    Push("Charmander"), Push("Squirtle"), Push("Bulbasaur"), Pop(), Push("Larvitar"), Push("Eevee"), Pop(), Pop()

    Which of the following is true after all the functions are executed?
    a. Charmander and Squirtle will be the only elements left on the stack
    b. The head node will be pointing to Charmander
    c. If you Push("Pikachu"), it will be the last node of the linked list

5) Which of the following is true about using the binary search algorithm in a sorted array with no holes and always keeping the order of the array intact?
    a. Binary search improves the performance of delete to O(log n)
    b. Binary search improves the performance of update to O(log n)
    c. Binary search improves the performance of search to O(log n)

6) Given an array [10 9 8 7 6 5 4 3 2 1], which of the following is true of the following sorting algorithms if we wanted to sort this array into ascending order?
    a. If we used optimized bubble sort on this array, the performance is O(n)
    b. If we used MergeSort on this array, the performance is O(n log n)
    c. If we used InsertionSort on this array, the performance is near O(n)

7) Which of the following is true about Big-O notation?
    a. If we say an algorithm performs at O(n), you will get the same number of operations executed no matter how large the data set grows in size
    b. If algorithm 1 is O(n) and algorithm 2 is O(1), algorithm 1 is considered to be significantly better than algorithm 2
    c. O(n log n) is significantly better than O(n)

8) Which of the following is true about the delete function in a standard linked list (only a head pointer and nodes only having a next pointer)?
    a. There is only 1 scenario to consider when looking for a node to delete in a linked list
    b. When you traverse the list to find the element to delete, it improves the performance to O(1)
    c. You should always check first to see if the list is empty before attempting to find and delete a node

9) Which of the following is true about the Mergesort algorithm applied to an array of numbers that are all different?
    a. Memory usage is efficient because all swaps take place within the same array
    b. Performance is consistently at O(n log n) for all array scenarios
    c. When the partition function completes on the array, the pivot element will always be in the exact center of the array

10) Which of the following is true about the linked list functions where the linked list only contains a head pointer?
    a. If the linked list is ordered, you can use binary search and improve the performance to O(log n)
    b. The worst case performance of the delete function is in the same Big-O category as the worst case performance of the delete function with an unordered array
    c. If the node you are deleting is at the end of the list, the performance is O(1)

Fun with Sorting (20 pts):

Given the following array of numbers:

8 2 3 9 10 1 4 6 7 5

Show what the array looks like after each iteration of the following sorting algorithms:

1) Bubble
2) Selection
3) Insertion
4) Mergesort

Only show the array contents with each algorithm. You do not need to show function call instances if recursion is used or write any code. Just show the array at key iterations of the algorithm. You can use your own words to describe them as well for more detail (but do not write any code).

Bonus (3 pts):

Show what the array looks like after each recursive iteration of the Quicksort algorithm.
You do not have to do this algorithm for the test. This is a bonus exercise

Code Analysis Section

Exercise 1 (15 pts)

Given the following recursive version of selection sort:

```
public void recursiveSelectionSort(int a[], int n, int index)
{
      if (index == n)
            return;

      int k = minIndex(a, index, n-1);

      if (k != index)
            swap(a, k, index);

      recursiveSelectionSort(a, n, index + 1);
}
```

minIndex is a separate function that finds the smallest value in the given array "a" from "index" to "n-1" index values.

swap is a separate function that swaps elements in the given array "a" between the elements at index "k" and "index" respectively.

Assuming these 2 functions work as expected, there may be an error with the recursiveSelectionSort code. Answer the following:

1) Design your own set of 8 unsorted integers in an array
2) Determine what the given code will produce with your array by describing what the array looks like with **each recursive instance of recursiveSelectionSort**
3) If the code does have an error, describe the error, where it is, and how you would fix it

For example, if you said your array was [4 2 3 1 5 6 7 8], then based on the code given here, you would show what the array would look like in the first recursiveSelectionSort instance and each of the subsequent instances like:

1st recursive instance: 4 2 3 1 5 6 7 8
2nd recursive instance: 2 3 4 5 1 6 7 8
3rd recursive instance: 2 3 4 5 1 6 7 8
Etc.

Then, you would use this to say whether or not the code as it is written is correct. If the given code is not correct, show the code change(s) you would make to fix it.

Exercise 2 (10 pts)

Given the following function to update a node in a standard linked list:

```
public boolean update(Node head, int oldValue, int newValue)
{
      if (head == null)
            return false;
      while (head != null)
            if (head.data == oldValue)
            {
                  head.data = newValue;
                  return true;
            }
            else
                  head = head.next;
      return false;
}
```

There may be an error with the code. Answer the following:

1) Design your own linked list of 4 nodes with different values
2) If the code has an error, describe the issue and provide an example of calling the update function with values of your choice and show what the linked list looks like
    a. Also, show the code changes you would make to fix the error
3) If the code does not have an error, provide an example of calling the update function with values of your choice and show what the linked list looks like

For example, if your linked list was 1 -> 2 -> 3 -> 4, and you think there is an error with this code, you could say something like "the problem with the code is xxx and after calling update(3, 5), the linked list will look like (and then show your linked list)"

If you think the code does not have an error, still provide an example of calling update with your own values and show what the linked list ends up looking like

Bonus section (7 pts)

Given the following sequence of numbers to insert into a tree data structure:

8 10 2 4 5 1 6 3 7 9

Answer the following questions:

1) What will the binary search tree look like after the numbers are inserted in the given order?
2) From the tree you finished in 1), what will the tree look like each time after the following numbers are deleted (use the delete algorithm discussed in the slides where the successor is found in the right child subtree):
    a. Delete(7)
    b. Delete(10)
    c. Delete(2)
3) From the tree you finished in 2), what will the preorder and postorder output traversals look like?

This is also a bonus section for the test. You do not have to do this section for the exam