CS 445
Quiz 2

Name: SOLUTION

Note: For True or False questions, correct the statement if it is False.

1. (5 points) What are the reasons someone should use Merge Sort instead of Quick Sort?

      1. Merge sort's theoretical worst-case runtime is better than quick sort's
      2. Merge sort is a stable sorting algorithm, quick sort is not

2. (7 points) What is the runtime of the following algorithm? Explain your answer. Assume that `ListInterface` is an interface to the List abstract data type discussed in class and that the `list` reference passed in is not null. Be sure to state and justify any assumptions you made.

```
public static void printValues(ListInterface<T> list)
{
    for (int i=1; i <= list.getLength(); i++)
    {
        System.out.println(list.getEntry(i));
    }
}
```

1. If the object refered to by list is an array-implemented List object, then O(n) because getEntry would be O(1) and the loop iterates n times.

2. If the object refered to by list is a linked list implemented List object, then $O(n^2)$ because getEntry would be O(n) and the loop iterates n times. The runtime for getEntry is O(n) because to get to value at position i, you must walk through the linked list each time starting at position 1.

The student must provide both to receive full credit.

3. (1 point) True or False: A circular linked list is a linked list where each node refers to both the next and the previous node in the linked list.

False. The student can correct it by saying that what's described is a doubly-linked list OR that a circular linked list is where the last node points back to the first node. They only need to provide one correction

4. (1 point) True or False: An iterator is a List object.

False. Corrected by either saying that an iterator is an Iterator object OR that an iterator is used to walk through the values of a List object OR by mentioning different List implementations (array or linked list)

5. (3 points) Why should a Linked List provide an iterator instead of having the programmer use the toArray method?

An iterator takes O(1) extra space while toArray takes O(n) extra space.

CS 445
Quiz 2

Name: SOLUTION

Note: For True or False questions, correct the statement if it is False.

1. (5 points) What are the reasons someone should use Insertion Sort instead of Quick Sort?

      1. For small arrays, Insertion sort is faster since there isn't the overhead of recursion
      2. For mostly-sorted arrays, insertion sort is much faster than quick sort

2. (7 points) What is the runtime of the following algorithm? Explain your answer. Assume that `ListInterface` is an interface to the List abstract data type discussed in class and that the `list` reference passed in is not null. Be sure to state and justify any assumptions you made.

```
public static void printValues(ListInterface<T> list)
{
    for (int i=1; i <= list.getLength(); i++)
    {
        System.out.println(list.getEntry(i));
    }
}
```

1. If the object refered to by list is an array-implemented List object, then O(n) because getEntry would be O(1) and the loop iterates n times.

2. If the object refered to by list is a linked list implemented List object, then $O(n^2)$ because getEntry would be O(n) and the loop iterates n times. The runtime for getEntry is O(n) because to get to value at position i, you must walk through the linked list each time starting at position 1.

The student must provide both to receive full credit.

3. (1 point) True or False: A doubly linked list is a linked list where each node holds two values instead of one.

False. A doubly linked list is a linked list where each node holds two references (one to the next node, one to the previous node) instead of one reference.

4. (1 point) True or False: An iterator is a List object.

False. Corrected by either saying that an iterator is an Iterator object OR that an iterator is used to walk through the values of a List object OR by mentioning different List implementations (array or linked list)

5. (3 points) Why should an array-implemented List provide an iterator instead of having the programmer use the getEntry method?

The student just needs to provide one of these:
1. To be consistent with the linked-list implemented List, which provides an iterator
2. As a convenient way of keeping track of where you left off in a list.
They can also say that there isn't a reason, as long as you find it convincing.