CSC345 Discussion 8

Going Over Quizzes and Practice Questions

Question 1 Quiz 3

Which of these is the best definition for a stable sorting algorithm?

A. An algorithm that always gives the same order for duplicate keys from run to run.
B. An algorithm that is as fast as the best one known
C. An algorithm that always gives the right answer
D. An algorithm that does not change the relative ordering of records with identical keys.

Question 1 Quiz 3

Which of these is the best definition for a stable sorting algorithm?

A. An algorithm that always gives the same order for duplicate keys from run to run.
B. An algorithm that is as fast as the best one known
C. An algorithm that always gives the right answer
D. An algorithm that does not change the relative ordering of records with identical keys.

Question 2 Quiz 3

Sometimes, the constant factors in an algorithm's runtime equation are more important than its growth rate. When the problem is sorting, this can happen in which situation?

A. When the amount of available space is small
B. When the records are nearly reverse sorted
C. When we are sorting lots of small groups of records
D. When the records are nearly sorted
E. When there are lots of records
F. When the CPU is really fast

Question 2 Quiz 3

Sometimes, the constant factors in an algorithm's runtime equation are more important than its growth rate. When the problem is sorting, this can happen in which situation?

A. When the amount of available space is small
B. When the records are nearly reverse sorted
C. When we are sorting lots of small groups of records
D. When the records are nearly sorted
E. When there are lots of records
F. When the CPU is really fast

Question 3 Quiz 3

What is the best case time for insertion sort to sort an array of n records?

Question 3 Quiz 3

What is the best case time for insertion sort to sort an array of n records?

$\Theta(n)$

This occurs when the keys begin in sorted order from lowest to highest. In this case, every pass through the inner for loop will fail immediately, and no values will be moved. The total number of comparisons will be $n - 1$, which is the number of times the outer for loop executes. Thus, the cost for Insertion Sort in the best case is $\Theta(n)$.

Question 4 Quiz 3

How many inversions are there in the following array?

688	851	460	259	65	897

Question 4 Quiz 3

How many inversions are there in the following array?

2 3 4

688	851	460	259	65	897

9 Total Inversions

The number of inversions (i.e., the number of values greater than a given value that occur prior to it in the array) will determine the number of comparisons and swaps that must take place.

Question 4 Quiz 3

What is the running time of Bubble Sort (as presented in class and text) when the input is an array where all the recorded values are equal?

Question 4 Quiz 3

What is the running time of Bubble Sort (as presented in class and text) when the input is an array where all the recorded values are equal?

$\Theta(n^2)$

From Shaffer:
Bubble Sort’s running time is roughly the same in the best, average, and worst cases.

Question 1 Quiz 4

1. In the average case, the total number of swaps done by Selection Sort is closest to:

A. n
B. n^2
C. n log n
D. n^2 /2

Question 1 Quiz 4

1. In the average case, the total number of swaps done by Selection Sort is closest to:

A. n
B. n^2
C. n log n
D. n^2 /2

Question 2 Quiz 4

Trace the changes of the following array in one iteration of the insertion sort. The current item being operated on is indicated by the arrow. Show every change in the array during the iteration.

241	244	628	767	886	657	40	819	673

Question 3 Quiz 4

3.log n! is:

$\Theta(n)$

$\Theta(\log n)$

$\Theta(n^2)$

$\Theta(n \log n)$

$\Theta(n^n)$

Question 3 Quiz 4

3.log n! is:

$\Theta(n)$

$\Theta(\log n)$

$\Theta(n^2)$

$\Theta(n \log n)$ Using Stirling's approximation

$\Theta(n^n)$

Question 4 Quiz 4

If the upper and lower bounds for a problem meet then:

A. The universe explodes
B. We are ready to implement the solution to the problem
C. We can say that we understand the runtime cost of the problem
D. The problem is $\Theta(n)$

Hint: The universe hasn't exploded yet.

Question 4 Quiz 4

If the upper and lower bounds for a problem meet then:

A. The universe explodes
B. We are ready to implement the solution to the problem
C. We can say that we understand the runtime cost of the problem
D. The problem is $\Theta(n)$

Just because the bounds meet, it does not mean that the cost is $\Theta(n)$.

We might need to implement an algorithm even if we are not certain that it is the best.

Question 5 Quiz 4

The order of the input records has what impact on the number of comparisons required by Bin Sort (aspresented in the book and class)?

A. None
B. There is a constant factor difference
C. There is a big difference, the asymptotic running time can change

Question 5 Quiz 4

The order of the input records has what impact on the number of comparisons required by Bin Sort (aspresented in the book and class)?

A. None
B. There is a constant factor difference
C. There is a big difference, the asymptotic running time can change

The behavior is the same for all input.

Question 6 Quiz 4

The order of the input records has what impact on the number of comparisons required by Merge Sort (as presented in the book and class)?

A. None
B. There is a constant factor difference
C. There is a big difference, the asymptotic running time can change

Question 6 Quiz 4

The order of the input records has what impact on the number of comparisons required by Merge Sort (as presented in the book and class)?

A. None
B. There is a constant factor difference
C. There is a big difference, the asymptotic running time can change

Question 7 Quiz 4

Insertion Sort (as presented in the book and class) is a stable sorting algorithm.

True

False

Question 7 Quiz 4

Insertion Sort (as presented in the book and class) is a stable sorting algorithm.

True

False

Practice Question

Write an algorithm to find a **maximum** cost spanning tree, that is, the spanning tree with highest possible cost.

Practice Question

Assume L is an array, L.length() returns the number of records in the array, and qsort(L, i, j) sorts the records of L from i to j (leaving the records sorted in L) using the Quicksort algorithm. What is the average case time complexity for each of the following code fragments?

(a)
```
for (i=0; i<L.length; i++)
        qsort(L, 0, i);
```

(b)
```
for (i=0; i<L.length; i++)
        qsort(L, 0, L.length-1);
```

Practice Question

Here is a variation on sorting. The problem is to sort a collection of n nuts and n bolts by size. It is assumed that for each bolt in the collection, there is a corresponding nut of the same size, but initially we do not know which nut goes with which bolt. The differences in size between two nuts or two bolts can be too small to see by eye, so you cannot rely on comparing the sizes of two nuts or two bolts directly. Instead, you can only compare the sizes of a nut and a bolt by attempting to screw one into the other (assume this comparison to be a constant time operation). This operation tells you that either the nut is bigger than the bolt, the bolt is bigger than the nut, or they are the same size. What is the minimum number of comparisons needed to sort the nuts and bolts in the worst case?