CS 2420
Fall semester section 01
Ricardo Gonzalez Mendez

Chapter 8 notes

Shell sort: subquadratic time
Merge sort: O(N log N)
Quicksort: O(N log N)

Big Omega is it will never be faster then this.

Sorting is important because,
Dictionary
Index
Card cataloged in library

For a duplicate if it is unsorted then the big Omega is quadratic
If It was sorted then then it would be Big Oh would be linear because then you just have to look at the one next to you

Insertion sort
Worst case: O(N^2)
Average case: O(N^2)
Best case: O(N)
Start with array of 1 (it is sorted)
Get an item
Run down the list until you find the correct place

With 5 people
With the Big Oh It would O(N^2)

Start insertion at 1 because 0 is already sorted.

Test question
For an array that is already sorted then insertion has a best case of run time linear.

Inversion
Is a pair of elements that are out of order in an array. In other words it is any ordered pair (I, j)

Theorem: average number of inversions in an array of N distinct numbers Is N(N-1)/4
Theorem: any algorithm that sorts by exchanging adjacent elements requires Big Omega(N^2) time on average.

Shell sort
Worst case: O(N Log N)
Average case: O(N Log N)
Best case:
This is smarter and faster but doesn't seem like it.
Sort by 5 positions
Then sort by 3 positions
Then sort by 1 positions

Merge Sort
Worst case: O(N Log N)
Average case: O(N Log N)
Best case: O(N log N)
Use recusion

Use recusion
Base case array of size 1 or 0
Make progress by sorting by halves of the array size is 5
Gotta believe by using your recusion
Wont have to worry about that last step.

Quick Sort
Worst case: O(N^2)
Average case: O(N log N)
Best case:
Pick any item in an array
Sort in into 3 groups example 50

- Smaller then group <50
- Equal group ==50
- bigger then group >50

For recusive
If S is 0 or 1 return base case
Picking the pivot
Partition into 2 disjointed groups
Quicksort left and right

Picking a pivot
Picking the first element is not a good idea, it would the worst choice, also picking the last element would be a bad choice
Middle element is safe since it is already sorted, this is the best choice.

Big O doesn't come into play when things are small

If you want to get faster when you are left with 10 things then just through insertion sort in there.

For the last question you put 10 numbers that are the worst for insertion\ sort