TARUSI MITTAL
1901CS65

Question 1 - Comment if marge sort, quick sort, insertion sort, heap sort, bucket sort are stable sort or not with proper justification.-

Answer:-

Merge Sort :- In merge sort, an array is recursively divided into halves.In subsequent steps during merging preference is given to left half while arranging the elements of sorted subarray. Here if two elements are equal of both the halves the element from left array is put into sorted array(at this point if right element would be chosen it will become unstable). Generally it is considered to be stable, but as seen it depends upon the merge function called in it.

Quick Sort: - As we do swap of elements according to pivot's position It is unstable.

Insertion Sort: - In Insertion sorting we just pick an element and place it in its correct place and in this way only if the element is larger than the key it will get swapped else it remains same i.e., we are not swapping the element with key when it holds equality condition. So, it a stable sorting.

Heap sort :- During the heap creation stage, any information about the ordering of the items in the original sequence gets lost. Heap sort is thus unstable.

Consider a max-heap -- 52 48(1) 48(2) 40 10 3

After heap-sort -- 3 10 40 48(2) 48(1) 52

It does not preserve the order of elements and hence can't be stable.

Bucket Sort :- Bucket sort uses insertion sort in its buckets to sort the data.Since insertion sort is stable bucket sort is also stable

Question 2- Comment what would happen if radix sorting is started from MSB.

Answer:-

In general radix sort can not be started from msb as it will arrange the numbers according to lower places ie. units place tens place etc..

for example(Arranging in descending order)

```
:- 124 -> 247 -> 247 -> 217 -> 217   (No proper order in the result)

   247 -> 217 -> 124 -> 247 -> 247

   217 -> 124 -> 217 -> 124 -> 124
```

However, Radix sort can be implemented in a way so as to start from MSB. Function can be made to do counting sort only on those digits of numbers whose previous digits were equal,else the numbers are already sorted