Lab-6

1. Show that any comparison-based algorithm to Sort 4 elements requires at least 5 comparisons in the worst case.

⇒ Every comparison based algorithm can be represented as decision tree. The labels on the links of the tree represent comparison items as the algorithm runs.

For 4 elements; Correct comparison-based sort must have at least n! leaves.

$4! = 4 \times 3 \times 2 \times 1 = 24$

The maximum no. of leaves for a full binary tree (Decision tree) of depth d is 2^d So, $2^5 = 32$ (because for 4 element height of tree $= 2^5$)

Hence, # comparison performed in worst case

= depth of the deepest node

$= 2^5 = 32$

∴ for 4 element there is at least 5 comparisons.

More: $\log 4! = \log 24 < \log 32 = \log 2^5 = 5$

③ Carry out the steps of RadixSort to sort the following {80, 27, 72, 1, 27, 8, 64, 34, 16} - Hint: use 9 for your radix.

<u>procedure:</u>

step1: scan initial array and make remainder bucket.

r[] =

27								8
72	64						16	~~72~~
27	1						34	80
0	1	2	3	4	5	6	7	8

e.g.

First, $80 \% 9 = 8$

Similarly calculate the remainder array.

Again,

quotient bucket,

q[] =

			34					
8			27					80
1	16		27				64	72
0	1	2	3	4	5	6	7	8

As above, calculate the quotient from the remainder bucket. for example:

$27/9 = \underline{\underline{3}}$, so position of 27 is index.

Finally,

1, 8, 16, 27, 27, 34, 64, 72, 80 is sorted result.