PAGE:
DATE: / /

* Assignment No. 6 *

Aim:-
write a python program to store first year percentage of student in array. write function for storing array of floating point numbers in ascending order using quick sort & display top 5 scores.

objectives:-
1. To understand the standard & abstract data representation methods.
2. to identify the appropriate data structure & algorithm design method for a specified application.
3. to get the sorted array of percentage of first year student using quick sort.
4. to get top 5 scores of student after quick sort.

outcomes:
1. to understand, design & implement quick sort using list or array in python.
2. to analyse the quick sort time complexity.
3. to design the algorithm to sort the programming problems.

o/s / programming tools used:
64-bit Fedora 17 or latest 64-bit update of equivalent open source os or latest 64 bit version of microsoft window 7. Eclipse with python plugin.

Page No.
Date

theory :-

quick sort :-

- Quick sort is also known as partition exchange sort based on the rule of divide and conquer.
- It is highly efficient sorting algorithm.
- Quick sort is the quickest comparison based sorting algorithm.
- It is very fast & requires less additional space, only o(nlogn) space reqursed.
- Quick Sort picks an element as pivot and partitions the array around the picked pivot.

- there are different version of quick sort which choose the pivot in different ways:
  1. first element as pivot.
  2. Last element as pivot.
  3. Random element as pivot.
  4. median as pivot.

Algorithm for quick sort:

step 1 : choose the highest value as pivot.
step 2 : Take two variable to point left and right of the list excluding pivot.
step 3 : left points to the low index.
step 4 : Right points to the high index.
step 5 : while value at left < pivot move right
step 6 : while value at right > pivot move left

Page No.
Date

step 7: If both steps 5 & step 6 does not match swap left and right.
Step 8: If left = right, the point where they met is new pivot.

Pseudo code for recursive Quick Sort function
/* How → starting index. high → Ending index */.

```
Quick sort ( arr [], low high ) {
 if ( low < high ) {
        pi = partition ( arr, low, high );
       quicksort ( arr, low, pi-1 );
        quicksort ( arr, pi+1, high )
  }
}.
```

Pseudo code for partition():

```
partition ( arr [] :, low, high ) {
      pivot = arr [high];
       i = (low-1);
for (j=low ; j <= high-1 , j++ ) {
   if ( arr [j] < pivot ) {
      i++;
   Swap arr [i] & arr [j].

   }
}.
```

(3,1) G(4)

Page No.
Date

Swap arr [i+1] & arr [high])
return (i+1)

time Complexity : o(n log n).
Space complexity : o(log n)

Conclusion :-
Quicksort implemented successfully for getting top 5 student.