**Assignment No 5**

**Problem Statement:**

Write a **Python** program to store first year percentage of students in array. Write function for sorting array of floating point numbers in ascending order using quick sort and display top five scores.

**Theory:**

|  |
| --- |
| Example of quick sort( Till first partition is made)  ../_images/firstsplit.png  ../_images/partitionA.png  ../_images/partitionB.png   Completing the Partition Process to Find the Split Point for 54  Same process will be repeated for left half and right half to sort whole array.  **Pseudo code for Quick sort:**   1. QUICKSORT (array A, **int** low, **int** high) 2. 1 **if** (high> low) 3. 2 p ← PARTITION (A, low, high) 4. 3.  QUICKSORT (A, low, p - 1) 5. 4. QUICKSORT (A, p + 1, high)  Pseudo code for Partition: Partition algorithm rearranges the sub arrays in a place.   1. PARTITION (array A, **int**  low**,  int** high) 2. 1 pivot ← A[low] 3. 2 i←low+1 4. 3. j←high 5. 4. Repeat step 5 and 6 till low<high 6. 5. while (pivot >A[i]) 7. Increment i 8. 6. while (pivot <A[j]) 9. decrement j 10. 7. if  i<j 11. swap A[i] with A[j] 12. 8 if(i>j) 13. swap A[j] with A[low] 14. 9 **return** j |
|  |
|  |

**Advantages of quicksort**

* The average-case time complexity to sort an array of n elements is O(n lg n).
* Generally, it runs very fast. It is even faster than merge sort.
* No extra storage is required

**Disadvantages of quicksort**

* Its running time can be different for different array contents.
* The worst-case quick sort takes place when the array is already sorted.
* It is not stable.

**Conclusion**

The speed of the quick sort algorithm makes it one of the most important sorting algorithms.