Assignment-5

Shrikrushna S Zirape
21286

Problem Statement / Defination:-

Write a python program to store second year percentage of student in array. Write function for sorting of array of floting point no's in ascending order using
a) insertion sort.
b) shell sort.
And display Top 5 scores.

Objective:-
- To understand Concept of Sorting
- To study different sorting method
- To study application of each method.

Outcome -
- To write function for implimenting searching method
- To use function written for only one application.

S/W and H/W Requirements:-

- programming languages - python
- operating system - 64 bit fedora.
- programming tool - Jupyter note book.

Theory:-

Insertion Sort:-

Insertion sort is a simple sorting algorithm that works similer to the way sort playing card in your hands. The array is virtually splits into a sorted and unsorted part. Values from unsorted part are picked and placed at the correct position in the sorted part.

Shell Sort:-

Shell sort mainly variation of insertion sort in insertion sort we move element only one position ahed many moments are involved the ideas of shell sort is allow exchange of far items we may take the array h-sorted for a large value of h we keep reducing value of h until it becomes 1

Algorithm:-

* main structure of program.

```
1   Start.
2   int t, ~~arr~~ r[          # t = total no of student
3   enter total no. of student
4    arr [t]
5    for i = φ  to  i = t
6        arr.append ( float(input))
7    end for
8   int Choice  , S1 = sort() --
9   if choice == 1
10        S1.insertionsort()
11  elif Choice == 2
12        S1.shellsort()
13  elif Choice == 3
14         break
15  else
16        print ("Wrong Choice")
17  end
```

* Function for length of array.

```
1   Start
2   take array Input , i = 0
3   append -1 to array
4   ~~for~~ while ~~dest~~ [i] != -1
5         i++
6   endloop
7   return i
8   stop
```

* Function for printing Top 5 Scores :

```
1   Start
2   take array input
3   ~~for~~ i=1
4   t=self. getlen(arr)
c   ~~i=6to~~   i =
5    for i=-1 to i=-6
6         print arr[i]
7   endfor
8    end
```

* Insertion Sort :-

```
1   declare  i, key , j
2   loop :- i=1  to  n-1
3        key = arr[i]
4        j = i-1
5        loop : (j>=0 && a[j]> key)
6             arr[j+1] = arr [j]
7             j -= 1
8        end loop
9        arr[j+1] = key
10  end loop
```

* Shell ~~sort~~ sort:-

```
1  Start.
2  input array
3  n = self.getlen(arr]
4  gap = n//2 ; temp=0
5  While (gap>0):
6        For i= 0 to gap
7            temp = lst[i]
8            while j >=gap && arr[j-gap] > temp
9            j ← j - gap
10       arr[j] = temp   # update arr[j]
11     gap = gap//2      # update gap
12 end
```

Time Complexity:-

insertion sort = ~~O(n*2)~~ O((n^2))
shell sort = ~~O(n*2)~~ O((n^2))

Test Case.	Description	i/p	o/p	exep output	result.
1	1. Insert^n Sort 2. Shell Sort. 3. exit.	5. [1.23, 98.64, 39.40, 90.32, 86.0] →1	[1.23, 39.40, 80.0, 90.32 98.54]	[1.23, 93.40, 86.0 90.32 98.64]	pass
2		6 [60.00, 70.00 80.00 90.00 95.00 91'00] →1	[60.00, 70.00, 80.00 90.00 91.00 95.00]	[60.00, 70.00, 80.00, 90.00, 91.00, 95.00]	pass.