21143 Tanmay Karmarkar

Insertion & Shell sort

Problem Statement:
Write a python program to store second year percentage of students in array. Write a function for sorting array in ascending order using
a) Insertion sort
b) Shell sort.

Objective: To:
Implement various sorting algorithms like insertion & shell sort.

Outcome: Student will be able to:
Write & execute python programme for sorting

S/W & H/W Requirements:
Windows 10, Python 3, AMD 5500U, 8 GB RAM, 1512 GB SSD.
VS CODE.

Theory: Sorting algorithms put elements of a list in a certain order. Most frequently used are- numerical orders.

Efficient sorting is important for optimizing efficiency of search.

Insertion sort is simple, efficient for small & mostly sorted lists. It works by taking element & inserts into right place. But is expensive & requires shifting of all elements.

Shell sort improves upon insertion sort by moving out of order elements more than one position at a time. First sorting element far away so that final sort is fast. For few out of place elements, it works faster.

21143 Tanmay Karmarkar Assignment-5.

Algorithm for insertion sort:

1. Start
2. Repeat till end of list.
   a. Set curr as element at pos. i in list.
   b. Repeat j as i-1.
   c. Repeat while j>0 & pos j-1 in list L is > than curr.
      i. Set element at pos j+1 in list L as element present at pos j in list L.
      ii. Decrement pos by 1.
   d. Set the ele. at pos j+1 as curr.
   e. Print list.
3. ~~Return~~ Stop

Algorithm for shell sort:

1. Start.
2. Set p as 0.
3. Set gap as length of list L floor division by 2.
4. Repeat the following while gap>0.
   a. Run a loop from 1 to gap to i equal to len(L).
   b. Run loop from j equal to i-gap to -1 with decrement as gap
      1. If ele at j+gap in L is > ele at j then exit from loop.
      2. Else
         swap elements.
   b. Set gap equal to floor division by 2.
   c. Increament p by 1.
   d. Print list.
5. Return list

Algo for main programme:

1. Start
2. Repeat.
3. Input from user the number of students
4. If less than 5 tell user to enter > 5 students.
5. Pay:
   Enter percentage of students and store in x.
   ii If $0 \leq x \leq 100$
   append to list.
   Else:
   print ("Not a valid percentage")
   Print ("Main menu")
   print (1. "insertion sort")
   print (2. "shell sort 3. exit")
   Take input & perform operation

Pseudocode

```
Class Sort {
def inssort (self, l).
for i in range (1, len(l)):
    curr = l[i]
    j = i-1
while (j>=0 & l[j] > curr]:
        l[j+1] = l[j]
        j = j-1
l[j+1] = curr.
  print (l)
return l.
```

```
def shell_sort (self, l)
{
p = 0
gap = len(l)//2
while (gap > 0)
for i in range (gap, len(l)):
  for j in range (i-gap, -1, -gap):
    if l[j+gap] > l[j]:
      break.
    else: l[j+gap], l[j] = l[j], l[j+gap]
gap = gap//2
p = p+1
print (l)
return l.

For taking input:
obj = sort().
while True:
  a = int (input)
  if a>5 : break
  else: print (a should be > than 5)

for i in range a:
  x = int (input())
  if 0 ≤ x ≤ 100:
    l.append (x)
    break.
  else: print (Not a valid percentage.
```