

Day 31

Sort an array of strings using selection sort

Input :

["GK", "annamalai", "arunsanthosh"]

Output :

["annamalai", "arunsanthosh", "GK"]

Implementation :

def selection-sort(a):

n = len(a)

for i in range(n):

min_idn = i

min-str = a[i]

for j in range(i+1, n):

~~#if~~ if min-str > a[j]:

min_idn = j

min-str = a[j]

if min_idn != i:

temp = a[i]

a[i] = a[min_idn],

a[min_idn] = temp

return a

Input:

a = ['gk', 'annamalai', 'arunsanthosh']

Selection-Sort(a)

Print(a)

Recursive bubble sort

class bubblesort :

def __init__(self, array):

self.array = array

self.length = length

def __str__(self):

return " ".join([str(n) for n in self.array])

def bubblesortRecursive(self, n=None):

if n is None:

n = self.length

if n == 1

return

for i in range(n-1):

if self.array[i] > self.array[i+1]:

~~self.array[i], self.array[i+1] = self.array[i],~~
~~self.array[i]~~

self.bubble sort recursive (n-1)

~~def~~ ~~un~~

Input:

def main():

array = [64, 34, 25, 12, 22, 11, 90]

sort = bubblesort(array)

sort.bubblesort recursive()

print(sort)

main()

)
Output:

[11, 12, 22, 25, 34, 64, 90]

Insertion Sort

insertion sort is a simple sorting algorithm.

The array is virtually split into a sorted and an unsorted part. Values from the unsorted part picked and placed at correct position in the sorted part.

Implementation :

def insertion-sort(a):

n = len(a)

for i in range(1, n):

key = a[i]

j = i - 1

while j >= 0 and key < a[j]:

a[j+1] = a[j]

j -= 1

a[j+1] = key

Input :

a = [24, 32, 41, 2, 4, 16]

insertion-sort(a)

print(a)

Output :

[2, 4, 16, 24, 32, 41]

Time Complexity :

$O(n^2)$