Lab sheet 2

Question 1 answer

Bubble Sort:

Algorithm:

* Loop through the array n-1 times.
* For each iteration, loop through the array from 0 to n-1-i.
* Compare the current element with the next element. If the current element is greater than the next element, swap them.
* After each iteration, the largest element is pushed to the end of the array.
* Repeat steps 2-4 until the array is sorted.

Psudo code

def bubble\_sort(array):

n = len(array)

for i in range(n-1):

for j in range(0, n-1-i):

if array[j] > array[j+1]:

array[j], array[j+1] = array[j+1], array[j]

return array

coding using python

def bubble\_sort(array):  
 n = len(array)  
 for i in range(n-1):  
 for j in range(0, n-1-i):  
 if array[j] > array[j+1]:  
 array[j], array[j+1] = array[j+1], array[j]  
 return array  
  
#sort the list of array  
array= [34, 67, 89, 34, 678, 12, 89]  
sorted\_array = bubble\_sort(array)  
print(sorted\_array)

def bubble\_sort(array):  
 n = len(array)  
 for i in range(n-1):  
   
 swapped = False  
 for j in range(0, n-1-i):  
 if array[j] > array[j+1]:  
 array[j], array[j+1] = array[j+1], array[j]  
 swapped = True  
   
 if not swapped:  
 break  
 return array  
  
# test the function  
array = [34, 67, 89, 34, 678, 12, 89]  
sorted\_array = bubble\_sort(array)  
print(sorted\_array)

b) Selection Sort:

Algorithm:

Loop through the array n-1 times.

For each iteration, find the minimum element in the unsorted part of the array.

Swap the minimum element with the first element in the unsorted part of the array.

Repeat steps 2-3 until the array is sorted.

Pseudo-code:

def selection\_sort(array):

n = len(array)

for i in range(n-1):

min\_idx = i

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

if array[j] < arayr[min\_idx]:

min\_idx = j

array[i], array[min\_idx] = array[min\_idx], arr[i]

return array

selection sort algorithem

psudo code;

for i in range(len(array)):

min\_idx = i

for j in range(i+1, len(array)):

if array[j] < array[min\_idx]:

min\_idx = j

array[i], array[min\_idx] = array[min\_idx], array[i]

code;

def selection\_sort(array):  
 for i in range(len(array)):  
 min\_idx = i  
 for j in range(i+1, len(array)):  
 if array[j] < array[min\_idx]:  
 min\_idx = j  
 array[i], array[min\_idx] = array[min\_idx], array[i]  
 return array

bucket sort;

psudo code;

def bucket\_sort(array, array\_size):

# Create an array

array1 = [[] for \_ in range(array\_size)]

for x in array:

index = int(x \* array\_size)

array1[index].append(x)

for i in range(array\_size):

array1[i].sort()

#sorted array assigning to the empty array

sorted\_array = []

for bucket in array:

sorted\_array += bucket

return sorted\_array

python code:

def bucket\_sort(array, array\_size):  
 # Create empty array1  
 array1 = [[] for \_ in range(array\_size)]  
  
 for x in array:  
 index = int(x \* array\_size)  
 array1[index].append(x)  
  
 for i in range(array\_size):  
 array1[i].sort()  
  
 # Concatenate the sorted buckets back into the original array  
 sorted\_array = []  
 for bucket in array1:  
 sorted\_array += bucket  
  
 return sorted\_array

selection sort;

def selection\_sort(array):  
 for i in range(len(array)):  
 min\_idx = i  
 for j in range(i+1, len(array)):  
 if array[j] < array[min\_idx]:  
 min\_idx = j  
 array[i], array[min\_idx] = array[min\_idx], array[i]  
 return array  
  
# Test the function  
array = [20, 46, 22, 19, 6, 42, 14, 5, 48, 47, 17, 39, 51, 7, 2]  
print("Original Array:", array)  
  
selection\_sorted\_array = selection\_sort(array)  
print("Selection Sort:", selection\_sorted\_array)