Kimberly D. Tandoc

IDB2

BSIT- 2nd Year

September 9, 2024

DSALGO1

ACTIVITY 1

Algorithm Implementation. Implement the algorithms indicated for each **data set**. **20 points for the implemented algorithm**.

1. **[23,89, 7, 56, 44]** – Implement the Bubble Sort Algorithm for the Dataset and sort the data into **ascending order**

```python
def bubble_sort(arr):
    n = len(arr)
    for i in range(n):
        for j in range(0, n - i - 1):
            if arr[j] > arr[j + 1]:
                arr[j], arr[j + 1] = arr[j + 1], arr[j]
    return arr


# Example usage
arr = [23, 89, 7, 56, 44]
print("Array before Bubble Sort",arr)



arr = bubble_sort(arr)
print("Array after Bubble Sort",arr)
print()
1 usage
```

OUTPUT:

```
Array before Bubble Sort [23, 89, 7, 56, 44]
Array after Bubble Sort [7, 23, 44, 56, 89]
```

2. **[12, 78, 91, 34, 62]** – Implement the Insertion Sort Algorithm for the Dataset and sort the data into **ascending order**.

```python
def insertion_sort(arr):
    for i in range(1, len(arr)):
        key = arr[i]
        j = i - 1
        while j >= 0 and key < arr[j]:
            arr[j + 1] = arr[j]
            j -= 1
        arr[j + 1] = key



# Example usage
arr = [12, 78, 91, 34, 62]
print("Array before Insertion Sort",arr)



insertion_sort(arr)
print("Array after Insertion Sort",arr)
print()
```

OUTPUT:

```
Array before Insertion Sort [12, 78, 91, 34, 62]
Array after Insertion Sort [12, 34, 62, 78, 91]
```

3. **[5, 99, 48, 15, 67]** – Implement the Selection Sort Algorithm for the Dataset and sort the data into **descending order**.

```python
def selection_sort_desc(arr):

    for i in range(len(arr)):
        max_idx = i
        for j in range(i + 1, len(arr)):
            if arr[max_idx] < arr[j]:
                max_idx = j
        arr[i], arr[max_idx] = arr[max_idx], arr[i]



# Example usage
arr = [5, 99, 48, 15, 67]
print("Array before Selection Sort",arr)

selection_sort_desc(arr)
print("Array after Selection Sort",arr)
print()

1 usage
```

OUTPUT:

```
Array before Selection Sort [5, 99, 48, 15, 67]
Array after Selection Sort [99, 67, 48, 15, 5]
```

4. **[38, 82, 25, 74, 13] – Implement the Insertion Sort Algorithm for the Dataset and sort the data into descending order.**

```python
def insertion_sort_desc(arr):

    for i in range(1, len(arr)):
        key = arr[i]
        j = i - 1
        while j >= 0 and key > arr[j]:
            arr[j + 1] = arr[j]
            j -= 1
        arr[j + 1] = key


# Example usage
arr = [38, 82, 25, 74, 13]
print("Array before Insertion Sort",arr)


insertion_sort_desc(arr)
print("Array after Insertion Sort",arr)
print()
```

OUTPUT:

```
Array before Insertion Sort [38, 82, 25, 74, 13]
Array after Insertion Sort [82, 74, 38, 25, 13]
```

5. **Copy** all of the values from the second index and third index of the previous datasets into one dataset. After copying, sort the data into ascending order and descending order each order of the dataset is inserted into a separate list/array.

```python
def sort_array(arr):

    arr_asc = sorted(arr)
    arr_desc = sorted(arr, reverse=True)
    return arr_asc, arr_desc


# Example usage
arr = [7, 56, 91, 34, 48, 15, 25, 74]
print("Array before sorting",arr)

arr_asc, arr_desc = sort_array(arr)
print("Array in ascending order",arr_asc)

print("Array in descending order",arr_desc)
print()
```

OUTPUT:

```
Array before sorting [7, 56, 91, 34, 48, 15, 25, 74]
Array in ascending order [7, 15, 25, 34, 48, 56, 74, 91]
Array in descending order [91, 74, 56, 48, 34, 25, 15, 7]
```

6. **Create a new list/array or values copying all of the values from <u>item number 1 to 4</u>. Implement the Selection Sort Algorithm and sort the data into ascending order.**

```python
def selection_sort(arr):
    arr = [23, 89, 7, 56, 44, 12, 78, 91, 34, 62, 5, 99, 48, 15, 67, 38, 82, 25, 74, 13]
    print("Array before Selection Sort",arr)


    for i in range(len(arr)):
        min_idx = i
        for j in range(i + 1, len(arr)):
            if arr[min_idx] > arr[j]:
                min_idx = j
        arr[i], arr[min_idx] = arr[min_idx], arr[i]
selection_sort(arr)
print("Array after Selection Sort",arr)
print()
```

OUTPUT:

```
Array before Selection Sort [7, 56, 91, 34, 48, 15, 25, 74]
Array after Selection Sort [7, 15, 25, 34, 48, 56, 74, 91]
```

7. **Print the even and odd values of the list/array created in item number 6.**

```python
def even_odd(arr):
    arr = [23, 89, 7, 56, 44, 12, 78, 91, 34, 62, 5, 99, 48, 15, 67, 38, 82, 25, 74, 13]

    even_values = [x for x in arr if x % 2 == 0]
    odd_values = [x for x in arr if x % 2 != 0]
even_odd(arr)
print("Even values:",even_values)


print("Odd values:",odd_values)
print()
```

OUTPUT:

```
Even values: [34, 48, 56, 74]
Odd values: [7, 15, 25, 91]
```