Algorithm Mini Project

Name:-Sanket Shivaji Thorat Roll.no.:-33

Class:-Msc CS part 1 Subject:-Algorithm

Academic Year 2021-22

INDEX:-

1.Merge Sort

Aim:-Write a Python program to implement Merge-Sort

FileName:-mergesort.py (Github-link)

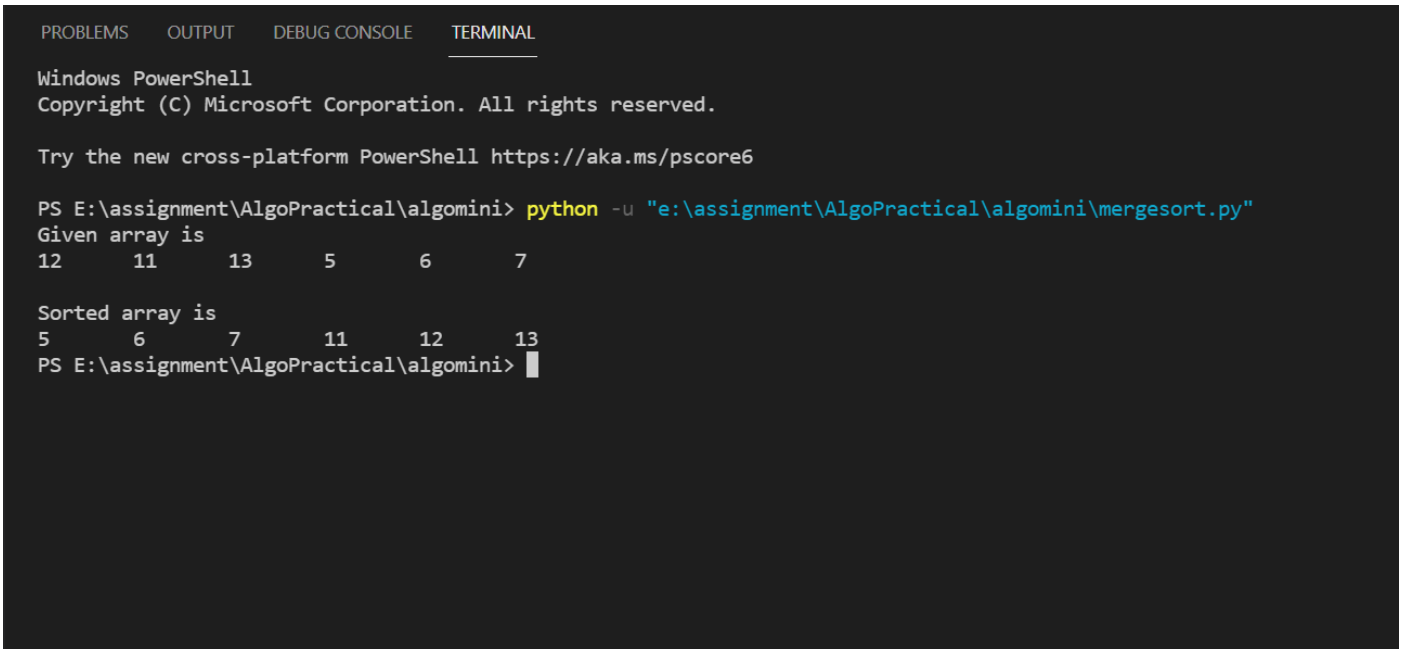
Code:-

```
# Python program for implementation of MergeSort


# Merges two subarrays of arr[].
# First subarray is arr[l..m]
# Second subarray is arr[m+1..r]


def merge(arr, l, m, r):
    n1 = m - l + 1
    n2 = r - m

    # create temp arrays
    L = [0] * (n1)
    R = [0] * (n2)

    # Copy data to temp arrays L[] and R[]
    for i in range(0, n1):
        L[i] = arr[l + i]

    for j in range(0, n2):
        R[j] = arr[m + 1 + j]

    # Merge the temp arrays back into arr[l..r]
```

```
    i = 0     # Initial index of first subarray
    j = 0     # Initial index of second subarray
    k = l     # Initial index of merged subarray

    while i < n1 and j < n2:
        if L[i] <= R[j]:
            arr[k] = L[i]
            i += 1
        else:
            arr[k] = R[j]
            j += 1
        k += 1

    # Copy the remaining elements of L[], if there
    # are any
    while i < n1:
        arr[k] = L[i]
        i += 1
        k += 1

    # Copy the remaining elements of R[], if there
    # are any
    while j < n2:
        arr[k] = R[j]
        j += 1
        k += 1

# l is for left index and r is right index of the
```

```
# sub-array of arr to be sorted


def mergeSort(arr, l, r):
    if l < r:

        # Same as (l+r)//2, but avoids overflow for
        # large l and h
        m = l+(r-l)//2

        # Sort first and second halves
        mergeSort(arr, l, m)
        mergeSort(arr, m+1, r)
        merge(arr, l, m, r)


# Driver code to test above
arr = [12, 11, 13, 5, 6, 7]
n = len(arr)
print("Given array is")
for i in range(n):
    print("%d" % arr[i]),

mergeSort(arr, 0, n-1)
print("\n\nSorted array is")
for i in range(n):
    print("%d" % arr[i]),
```

output:-


PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.
Try the new cross-platform PowerShell https://aka.ms/pscore6
PS E:\assignment\AlgoPractical\algomini> python -u "e:\assignment\AlgoPractical\algomini\mergesort.py"
Given array is
12 11 13 5 6 7
Sorted array is
5 6 7 11 12 13
PS E:\assignment\AlgoPractical\algomini>

2.Maximum SubArray Problem

Aim:- Write a Python program to implement the maximum subarray problem.

FileName:-MaxSubArray.py (Github-link)

Code:-

```
# Function to find the maximum sum of a subarray
# in a given integer array
def kadane(A):

    # stores the maximum sum sublist found so far
    max_so_far = 0

    # stores the maximum sum of sublist ending at the current position
    max_ending_here = 0

    # traverse the given list
    for i in A:
        # update the maximum sum of sublist "ending" at index `i` (by adding the
        # current element to maximum sum ending at previous index `i-1`)
        max_ending_here = max_ending_here + i

        # if the maximum sum is negative, set it to 0 (which represents
        # an empty sublist)
        max_ending_here = max(max_ending_here, 0)

        # update the result if the current sublist sum is found to be greater
        max_so_far = max(max_so_far, max_ending_here)

    return max_so_far
```

```
if __name__ == '__main__':

    A = [-2, 1, -3, 4, -1, 2, 1, -5, 4]

    print("The sum of subarray with the largest sum is",
        kadane(A))
```

Output:-