Algorithm Mini Project

Index

Maximum Subarray Problem

Aim: Write a Python3 program to implement the maximum subarray problem.

Filename: Maximum_Subarray_Problem.py

Code:

```
def maxSubArraySum(arr,size):
    max_till_now = arr[0]
    max_ending = 0
    for i in range(0, size):
        max_ending = max_ending + arr[i]
        if max_ending < 0:
            max_ending = 0
        elif (max_till_now < max_ending):
            max_till_now = max_ending

    return max_till_now

arr = [-2, -3, 4, -1, -2, 5, -3]
print("Maximum Sub Array Sum Is" , maxSubArraySum(arr,len(arr)))
```

Output:

```
Maximum Sub Array Sum Is 6
```

Quicksort

Aim: **Write a Python3 program to implement quick sort.**

Filename: Quicksort.py

Code:

```
def partition(arr, low, high):
    i = (low-1)         # index of smaller element
    pivot = arr[high]   # pivot
    for j in range(low, high):
        # If current element is smaller than or equal to pivot
        if arr[j] <= pivot:
            # increment index of smaller element
            i = i+1
            arr[i], arr[j] = arr[j], arr[i]
    arr[i+1], arr[high] = arr[high], arr[i+1]
    return (i+1)

def quickSort(arr, low, high):
    if len(arr) == 1:
        return arr
    if low < high:
        # pi is partitioning index, arr[p] is now at right place
        pi = partition(arr, low, high)
        # Separately sort elements before partition and after partition
        quickSort(arr, low, pi-1)
        quickSort(arr, pi+1, high)

# Driver code to test above
arr = [10, 7, 8, 9, 1, 5]
```

```
n = len(arr)
quickSort(arr, 0, n-1)
print("Sorted array is:")
for i in range(n):
        print("%d" % arr[i])
```

Output:

```
Sorted array is:
1
5
7
8
9
10
```