Algorithm Mini Project

Index

Maximum Subarray Problem

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

Filename: maximum_subarray_problem.py

Code:

```
from math import inf

def max_crossing_sublist(input_list, low, mid, high):
    left_sum = -1 * inf
    right_sum = -1 * inf
    sum = 0
    max_left = -1 * inf
    max_right = -1 * inf

    for i in range(mid, low - 1, -1):
        sum += input_list[i]
        if sum > left_sum:
            max_left = i
            left_sum = sum

    sum = 0
    for j in range(mid + 1, high + 1):
        sum += input_list[j]
        if sum > right_sum:
            max_right = j
            right_sum = sum

    return max_left, max_right, left_sum + right_sum


def max_subarray_problem(input_list, low, high):
    if low == high:
        return low, high, input_list[low]
    else:
        mid = int((low + high) / 2)
```

```
        left_low, left_high, left_sum =
max_subarray_problem(input_list, low, mid)
        right_low, right_high, right_sum =
max_subarray_problem(input_list, mid +  1, high)
        cross_low, cross_high, cross_sum =
max_crossing_sublist(input_list, low, mid, high)

        if left_sum >= right_sum and left_sum >= cross_sum:
            return left_low, left_high, left_sum
        elif right_sum >= left_sum and right_sum >= cross_sum:
            return right_low, right_high, right_sum
        else:
            return cross_low, cross_high, cross_sum


if __name__ == "__main__":
    from random import randrange
    input_list = [randrange(-100, 100) for i in range(20)]
    print("Input array is >", input_list)

    low_index, high_index, max_sum =
max_subarray_problem(input_list, 0, len(input_list) - 1)
    print("Maximum sum is: ", max_sum, "with ", low_index, "and",
high_index, 'as the lower and upper bounds.')
```

Output:

```
C:\Users\YASH\Documents\Practicals\Semester 1\Algorithms\mini_project>python maximum_subarray_problem.py
Input array is > [-65, -3, 64, -12, -37, 91, 26, 6, 6, -33, -64, -68, 56, -14, 21, 27, -21, 43, 87, 1]
Maximum sum is:  200 with  12 and 19 as the lower and upper bounds.
```

Quick Sort

Aim: Write a Python3 program to implement quick sort.

Filename: quick_sort.py

Code:

```
def partition(start_index, end_index, input_list):

    pivot_index = start_index
    pivot_value = input_list[pivot_index]

    while start_index < end_index:
        while start_index < len(input_list) and
input_list[start_index] <= pivot_value:
            start_index += 1

        while input_list[end_index] > pivot_value:
            end_index -= 1

        if start_index < end_index:
            input_list[start_index], input_list[end_index] =
input_list[end_index], input_list[start_index]

    input_list[end_index], input_list[pivot_index] =
input_list[pivot_index], input_list[end_index]

    return end_index

def quicksort(start_index, end_index, input_list):

    if start_index < end_index:
        p = partition(start_index, end_index, input_list)

        quicksort(start_index, p - 1, input_list)
        quicksort(p + 1, end_index, input_list)

if __name__ == "__main__":
    from random import randrange
```

```
input_list = [randrange(10, 100) for i in range(50)]
print("Input list is", input_list)
print()
quicksort(0, len(input_list) - 1, input_list)
print("Sorted list is", input_list)
```

Output:

```
C:\Users\YASH\Documents\Practicals\Semester 1\Algorithms\mini_project>python quick_sort.py
Input list is [71, 31, 70, 93, 49, 53, 95, 81, 15, 34, 68, 29, 89, 26, 39, 74, 73, 82, 11, 91, 43, 97, 23, 42, 73, 95, 88,
 62, 80, 22, 45, 64, 76, 41, 30, 61, 53, 43, 46, 29, 56, 14, 29, 23, 43, 70, 13, 57, 41, 49]

Sorted list is [11, 13, 14, 15, 22, 23, 23, 26, 29, 29, 29, 30, 31, 34, 39, 41, 41, 42, 43, 43, 43, 45, 46, 49, 49, 53, 53
, 56, 57, 61, 62, 64, 68, 70, 70, 71, 73, 73, 74, 76, 80, 81, 82, 88, 89, 91, 93, 95, 95, 97]
```