

Question 1.

```

1 def calculation_sum_between(A, low, mid, high) -> tuple:
2     def apply_sum_find_max(A, lower_bound, higher_bound, reverse=False) -> tuple:
3         holding_sum, max_sum, max_index = 0, -1e9, lower_bound
4         list_range = list(range(lower_bound, higher_bound))
5         if reverse:
6             list_range = list_range[-1::-1]
7
8         for i in list_range:
9             holding_sum += A[i]
10            if holding_sum > max_sum:
11                max_sum, max_index = holding_sum, i
12
13        return max_sum, max_index
14    left_max_sum, left_index = apply_sum_find_max(A, low, mid+1, reverse=True)
15    right_max_sum, right_index = apply_sum_find_max(A, mid+1, high+1)
16
17    return (left_index, right_index, left_max_sum + right_max_sum)
18
19 def max_sum_subarray(A, low, high):
20     if low == high:
21         return (A[low], low, high)
22
23     mid = int((low + high) / 2)
24     (left_sum, left_low, left_high) = max_sum_subarray(A, low, mid)
25     (right_sum, right_low, right_high) = max_sum_subarray(A, mid+1, high)
26     (left_between_index, right_between_index, between_sum) = calculation_sum_between(A, low, mid, high)
27     print(A[low:high+1], "left_sum is: %d right_sum: %d between_sum: %d" % (left_sum, right_sum, between_sum))
28     if right_sum >= left_sum and right_sum >= between_sum:
29         return (right_sum, right_low, right_high)
30     elif left_sum >= right_sum and left_sum >= between_sum:
31         return (left_sum, left_low, left_high)
32     else:
33         return (between_sum, left_between_index, right_between_index)
34
35 A = [4, -1, 2, 1, -3, -5, 4]
36 sum, origin, destination = max_sum_subarray(A, 0, len(A) - 1)
37 print("maximum sum: %d -> from: %d to: %d" % (sum, origin, destination))
    
```

TERMINAL

```

[4, -1] left_sum is: 4 right_sum: -1 between_sum: 3
[2, 1] left_sum is: 2 right_sum: 1 between_sum: 3
[4, -1, 2, 1] left_sum is: 4 right_sum: 3 between_sum: 6
[1, -3] left_sum is: 1 right_sum: -3 between_sum: -2
[-5, 4] left_sum is: -5 right_sum: 4 between_sum: -1
[1, -3, -5, 4] left_sum is: 1 right_sum: 4 between_sum: -3
[4, -1, 2, 1, -3, -5, 4] left_sum is: 6 right_sum: 4 between_sum: 7
maximum sum: 7 -> from: 0 to: 4
    
```

Question 2.

The first matrices are

$$\begin{aligned}
 S_1 &= 6 & S_6 &= 8 \\
 S_2 &= 4 & S_7 &= -2 \\
 S_3 &= 12 & S_8 &= 6 \\
 S_4 &= -2 & S_9 &= -6 \\
 S_5 &= 6 & S_{10} &= 14.
 \end{aligned}$$

The products are

$$\begin{aligned}
 P_1 &= 1 \cdot 6 = 6 \\
 P_2 &= 4 \cdot 2 = 8 \\
 P_3 &= 6 \cdot 12 = 72 \\
 P_4 &= -2 \cdot 5 = -10 \\
 P_5 &= 6 \cdot 8 = 48 \\
 P_6 &= -2 \cdot 6 = -12 \\
 P_7 &= -6 \cdot 14 = -84.
 \end{aligned}$$

The four matrices are

$$\begin{aligned}
 C_{11} &= 48 + (-10) - 8 + (-12) = 18 \\
 C_{12} &= 6 + 8 = 14 \\
 C_{21} &= 72 + (-10) = 62 \\
 C_{22} &= 48 + 6 - 72 - (-84) = 66.
 \end{aligned}$$

The result is

$$\begin{pmatrix} 18 & 14 \\ 62 & 66 \end{pmatrix}.$$

Question 3.

- Master Theory :

- $T(n) = a \cdot T(n/b) + O(n^d)$

a. $T(n) = 3T(n/9) + \sqrt{n}$

$$A = 3, b = 9, d = \frac{1}{2} \rightarrow a = b^d \rightarrow O(n^{1/2} \cdot \log(n))$$

b. $T(n) = T(n-4) + n$

$$\rightarrow T(n) = n + (n-4) + (n-8) + \dots + 4 + T(0)$$

$$T(n) = \frac{n}{8} [2a + (\frac{n}{4} - 1) \cdot 4] + T(0) = \frac{n}{8} [n + 4] + T(0) = \frac{n^2}{8} + \frac{n}{2} + T(0) \in$$

c. $T(n) = 6T(n/4) + n^2$

$$A = 6, b = 4, d = 2 \rightarrow a < b^d \rightarrow O(n^2)$$

d. $T(n) = 5T(n/2) + n^2$

$$A = 5, b = 2, d = 2 \rightarrow a > b^d \rightarrow O(n^{\log_2 5})$$

Question 4.

$$T(n) = 2T(n/3) + n \text{ for } n \geq 5$$

$$A = 2, b = 3, d = 1 \rightarrow b^d > a \rightarrow O(n)$$

Question 5.

We can get to correct answer by testing the options and also we can solve it by writing characteristic equation:

$$x^2 - 5x + 6 = 0 \rightarrow x = 2, 3 \rightarrow g(n) = \alpha_1(3)^n + \alpha_2(2)^n$$

Question 6.

Step 1 : moving n-1 discs from A to B using C

Step 2: moving one disc from A to C

Step 3: moving n-1 discs from B to A using C

Step 4: moving one disc from C to B

Step 5: moving n-1 disc from A to B using C

In Conclusion :

$$T(n-1) + 1 + T(n-1) + 1 + T(n-1) = 3T(n-1) + 2$$