#### Question 1.

•		
Д	EXPLORER ····	🔶 main.py 🔹 humps.py x 🔹initpy 🗅 🗘 🛄 …
	$\sim$ OPEN EDITORS	le humps,py > ♂ calculation_sum_between
0	🝨 main.py project_1_p M	1 def calculation_sum_between(A, low, mid, high) -> tuple:
	X 🍦 humps.py	2 def apply_sum_find_max(A, lower_bound, higher_bound, reverse=False) -> tuple:
	initpy project M	3 holding_sum, max_sum, max_index = 0, -1e9, lower_bound
9 o 013	V NEURAL_NETWORKS	4 list_range = list(range(lower_bound, higher_bound))
	- HEORAC_HEIMORKS	5 if reverse: 6 list range = list range[-1::-1]
	✓ neural_network	0 Lisciality - Lis
d B		. for i in list_range:
	✓ activation	9 holding_sum += A[i]
-0	>pycache	10 if holding_sum > max_sum:
	🗢initpy M	11 max_sum, max_index = holding_sum, i
	✓ layer	
₿		13 return max_sum, max_index 14 left max sum, left index = apply sum find max(A, low, mid+1, reverse=True)
	<pre></pre>	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)
ي الله	> metrics	i i i i i i i i i i i i i i i i i i i
		17 return (left_index, right_index, left_max_sum + right_max_sum)
	+ preprocess	18
1		19 def max_sum_subarray(A, low, high):
	🕹initpy 🛛 A	20 if low == high:
	> scheduling •	21 return (A[low], low, high)
	✓ train	22 23 mid = int((low + high) / 2)
-		23 made = antropy (w) + and (b) +
0	💠initpy 🛛 M	25 (right_sum, right_high) = max_sum_subarray(A, m, id+1, high)
	✓ utility_functions	26 (left_between_index, right_between_index, between_sum) = calculation_sum_between(A, low, mid, high)
	>pycache	<pre>27 print(A[low:high+1], "left_sum is: %d right_sum: %d between_sum: %d"% (left_sum, right_sum, between_sum))</pre>
	🗢initpy M	<pre>28 if right_sum &gt;= left_sum and right_sum &gt;= between_sum:</pre>
	🔹 init .pv 🛛 A	<pre>29 return (right_sum, right_tow, right_high) 30 elif left sum &gt;= right sum and left sum &gt;= between sum:</pre>
۲	<ul> <li>gitignore</li> </ul>	30 elif left_sum >= right_sum and left_sum >= between_sum: 31 return (left_sum, left_low, left_high)
	🔮 main.py M	31 reconnictericsum, tericitagn/ 32 else:
Ê	<pre>initpy</pre>	33 return (between_sum, left_between_index, right_between_index)
0		34
_	csp.py	35 A = [4, -1, 2, 1, 1, -3, -5, 4]
	humps.py	36 sum, origin, destination = max_sum_subarray(A, 0, len(A) - 1)
	🔮 one.py	<pre>37 print("maximum sum: %d -&gt; from: %d to: %d" % (sum, origin, destination))</pre>
_	🕏 t.py	TERMINAL SQL CONSOLE: MESSAGES DEBUG CONSOLE PROBLEMS OUTPUT 1: zsh 🗸 + 🖽 🛍 ^ ×
	Untitled.ipynb	[4, -]] 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 [3: 4 right_sum: 3 between_sum: 6 [1, -3] Left_sum [3: 1 right_sum: -2
	> TIMELINE	[-5, 4] left_sum is: -5 right_sum: 4 between_sum: -1
503	> NPM SCRIPTS	[1, -3, -5, 4] Left_sum is: 1 right_sum: 4 between_sum: -3 [4, -1, 2, 1, 1, -3, -5, 4] Left_sum is: 6 right_sum: 4 between_sum: 7
- MI	> MYSQL	$[i_1, -i_1, 2, j_1, j_2, -3, -3]$ (if could list or right sum: 4 between sum: 7 and from 18 to right sum 1 and from 18 to t 4
× 1	🖗 master*+ 🐵 Python 3.8.5 64-bi	it ('env') 🛞 0 🏠 0 🎧 🗇 Connect 📌 Live Share 🗇 CMake: [Debug]: Ready 🎇 No Kit Selected 🚳 Build [all] 🚓 🕨 Spaces: 4. UTF-8. L.F. Python 🖷 Go Live 🖉 🚨

## Question 2.

The first matrices are

The products are

$$S_1 = 6$$
  $S_6 = 8$   
 $S_2 = 4$   $S_7 = -2$   
 $S_3 = 12$   $S_8 = 6$   
 $S_4 = -2$   $S_9 = -6$   
 $S_5 = 6$   $S_{10} = 14$   
 $P_1 = 1 \cdot 6 = 6$   
 $P_2 = 4 \cdot 2 = 8$   
 $P_3 = 6 \cdot 12 = 72$   
 $P_4 = -2 \cdot 5 = -10$   
 $P_5 = 6 \cdot 8 = 48$ 

The four matrices are

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

 $P_6 = -2 \cdot 6 = -12 \ P_7 = -6 \cdot 14 = -84.$ 

The result is

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

## Question 3.

Master Theory :

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

- c.  $T(n) = 6T(n/4) + n^2$ A = 6, b = 4, d = 2  $\rightarrow$  a < b ^ d  $\rightarrow$  0(n ^ 2)
- d.  $T(n) = 5T(n/2) + n^2$ A = 5, b = 2, d = 2  $\rightarrow$  a > b ^ d  $\rightarrow O(n^{Log 5 base 2})$

# **Question 4**.

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

$$x^{2} - 5x + 6 = 0 \rightarrow x = 2, 3 \rightarrow g(n) = \alpha 1(3)n + \alpha 2(2)$$

# 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