SEMINAR 2

LISTS AND 2D LISTS

WEEK 6 – Seminar 2

Exercise 1:

Given list A of n numbers, find the largest possible sum of a contiguous sub-list.

-2	**1**	**-3**	**4**	**-1**	**2**	**1**	**-5**	**4**

For example, for the list given above, the contiguous sub-list with the largest sum is [4, -1, 2, 1], with sum 6 (in red).

Brute force approach

One obvious solution is to calculate the sum of every possible sub-list and the maximum of those would be the solution. We can start from index 0 and calculate the sum of every possible sub-list starting with the element A[0]. Then, we would calculate the sum of every possible sub-list starting with A[1], A[2] and so on up to A[n-1], where n denotes the size of the list. Note that every single element is a sub-list itself.

Using the pseudo code notation shown in Seminar 1, write the brute force algorithm to solve the problem. Given a list of size n, how many comparisons are done?

Could you think of a better approach?

This is quite difficult, and a better approach will be explained during the seminar session. But it does not hurt to have a thought about it beforehand.

Exercise 2:

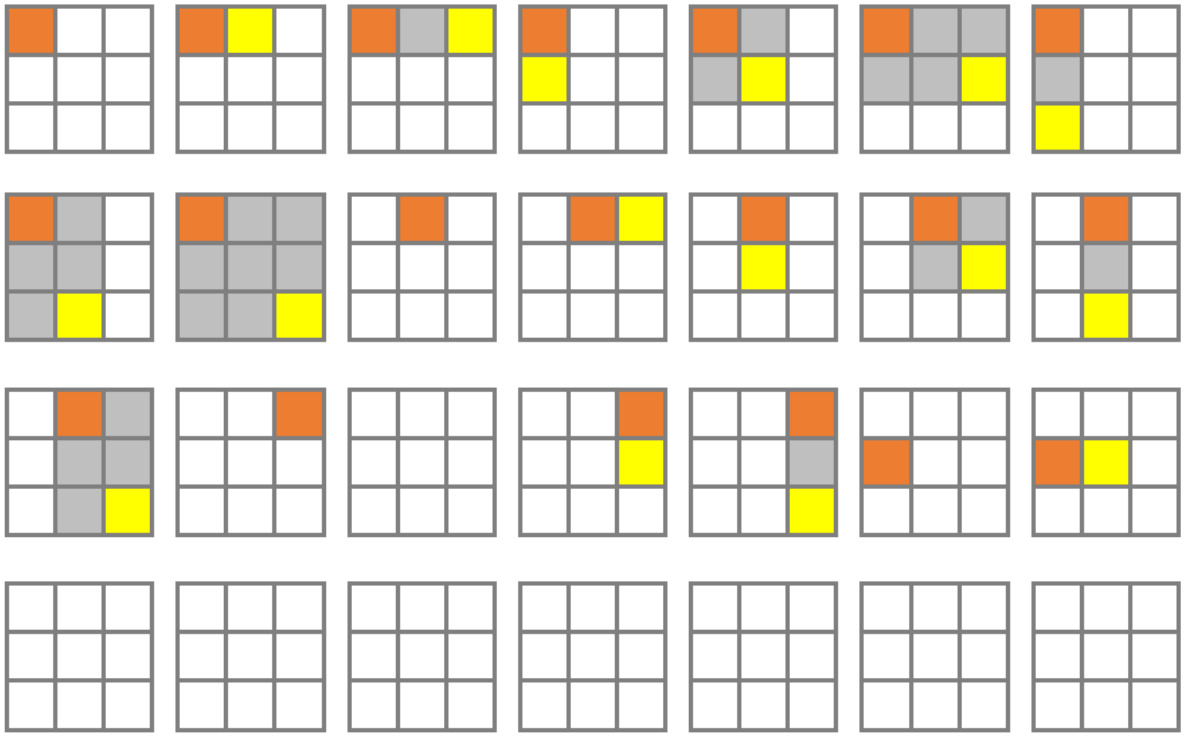
Given a matrix (2D list) $A[0..n-1, 0..m-1]$ of numbers, find the largest possible sum of a rectangle submatrix. The top-left element in the matrix is $A[0][0]$, and the element $A[r][c]$ is the element at row r, column c.

6	**-5**	**-7**	**4**	**-4**
-9	**3**	**-6**	**5**	**2**
-10	**4**	**7**	**-6**	**3**
-8	**9**	**-3**	**3**	**-7**

For example, for the matrix given above, the rectangle with the largest sum is highlighted in red with sum 17.

Brute force approach

The main idea behind the brute force algorithm is to try all possible rectangles within the 2D array. We can think of two cells to define a rectangle, the top-left corner (in orange) and the bottom-right corner (in yellow if not the same as top-left corner). Below you can find a subset of all the possible rectangle, and to understand better how to cover all possible rectangle, complete the bottom row of the image.



Once you have understood the principle, write the algorithm to find the maximal rectangle sum.