Solution to the Problem 6.4: Maximal Sum Square

Main Idea

Given an integer matrix $A[1..n][1..n]$ and number k. We are to find the $k \times k$ submatrix with maximal sum of its elements.

There are only $O(n^2)$ such submatrices, so we could go over all of them. But if we compute the sum of all elements of each submatrix naively, it will take $O(n^2 \cdot k^2)$ time. More accurately, this solution requires $(n-k+1)^2 k^2$ operations, and, for example, in case of $k = \frac{n}{2}$ the number of operations will be roughly equal to $\frac{n^4}{16}$. When $n = 700$, this does not fit into the time limit.

The idea of a possible correct solution is that with the help of a precalculation for $O(n^2)$ operations, we can get the sum of the elements of every submatrix in for $O(1)$ operations.

Let

$$\mathit{prefixSum}[i][j] = \sum_{k=1}^{i} \sum_{l=1}^{j} A[k][l] \tag{1}$$

for each $0 \le i, j \le n$. When i or j equals 0, we set $\mathit{prefixSum}[i][j]$ to 0.

The point is that for any submatrix we can express the sum of its elements through a combination of elements of the *prefixSum* matrix. Let $S[i_1][i_2][j_1][j_2]$ be the sum of all elements of a matrix $A[i_1..i_2][j_1..j_2]$. We claim that

$$\begin{aligned} S[i_1][i_2][j_1][j_2] = \mathit{prefixSum}[i_2][j_2] - \mathit{prefixSum}[i_2][j_1-1] - \\ - \mathit{prefixSum}[i_1-1][j_2] + \mathit{prefixSum}[i_1-1][j_1-1]. \end{aligned} \tag{2}$$

Let's prove it:

$$\begin{aligned} S[i_1][i_2][j_1][j_2] &= \sum_{k=i_1}^{i_2} \sum_{l=j_1}^{j_2} A[k][l] \\ &= \sum_{k=i_1}^{i_2} \left(\sum_{l=1}^{j_2} A[k][l] - \sum_{l=1}^{j_1-1} A[k][l] \right) \\ &= \sum_{k=1}^{i_2} \left(\sum_{l=1}^{j_2} A[k][l] - \sum_{l=1}^{j_1-1} A[k][l] \right) - \sum_{k=1}^{i_1-1} \left(\sum_{l=1}^{j_2} A[k][l] - \sum_{l=1}^{j_1-1} A[k][l] \right) \\ &= \sum_{k=1}^{i_2} \sum_{l=1}^{j_2} A[k][l] - \sum_{k=1}^{i_2} \sum_{l=1}^{j_1-1} A[k][l] - \sum_{k=1}^{i_1-1} \sum_{l=1}^{j_2} A[k][l] - \sum_{k=1}^{i_1-1} \sum_{l=1}^{j_1-1} A[k][l] \\ &= \mathit{prefixSum}[i_2][j_2] - \mathit{prefixSum}[i_2][j_1-1] - \\ &- \mathit{prefixSum}[i_1-1][j_2] + \mathit{prefixSum}[i_1-1][j_1-1]. \end{aligned} \tag{3}$$

It means that if we compute *prefixSum* matrix for $O(n^2)$ operations, we will be able to compute the sum of all elements of each $k \times k$ submatrix for $O(1)$ time, so we will reduce the complexity of our solution to $O(n^2)$.

We can do this using the following formula:

$$
\begin{aligned}
prefixSum[i][j] = prefixSum[i-1][j] + prefixSum[i][j-1] - \\
- prefixSum[i-1][j-1] + A[i][j].
\end{aligned}
\tag{4}
$$

It can be proven in the same way as the formula (2).

Implementation Details

It is most convenient to compute matrix *prefixSum* row by row, column by column:

```
for i from 1 to n:
  for j from 1 to n:
    prefixSum[i][j] ← prefixSum[i−1][j] + prefixSum[i][j−1] − prefixSum[i−1][j−1] + A[i][j]
```

Then we iterate over all $k \times k$ submatrices and compute the answer:

```
answer ← 0
for i from 0 to n−k:
  for j from 0 to n−k:
    answer ← max(answer, prefixSum[i+k][j+k] − prefixSum[i+k][j] − prefixSum[i][j+k] + prefixSum[i][j])
```