Problem 1

(Siegel Ex. 11.39). Let $A[1 \ldots n, 1 \ldots n]$ be an $n \times n$ array of signed numbers. Find the rectangular subarray with the largest sum of elements. A trivial solution is just to consider all $\Theta(n^4)$ rectangular subarrays; your algorithm should be much faster.

For example, if $A$ is the matrix

$$
\begin{bmatrix}
1 & 6 & -8 & 3 \\
3 & -10 & 4 & 5 \\
-2 & 2 & -1 & 6 \\
1 & 1 & 7 & -1
\end{bmatrix}
$$

the solution is

$$
A[2-4, 3-4] = \begin{bmatrix}
4 & 5 \\
-1 & 6 \\
7 & -1
\end{bmatrix}
$$

Problem 2

Let $T$ be a binary tree whose leaves are labelled with numbers. For any internal node $N$ of $T$, we define the imbalance of $N$ to be the absolute value of the difference between the sum of values in the left subtree and the sum in the right tree. Define the overall imbalance of $T$ to be the maximum imbalance of all internal nodes of $T$.

Given a sequence of numbers $S$, the most balanced tree for $S$ is the tree whose leaves are $S$ in the specified order with the smallest overall imbalance.

Write a dynamic programming algorithm that computes the most balanced tree for any sequence $S$. Hint: Use two arrays. $S[i, j]$ is the sum of elements $i$ through $j$. $B[i, j]$ is the overall imbalance of the best subtree spanning elements $i$ through $j$.

For instance, for the sequence $S = 4, 3, 1, 7$ there are five possible trees. The picture below shows the tree, with each node labelled with its sum $S$ and its imbalance $I$. The most balanced tree is $T_2$, with an overall imbalance of 2.
T1. Overall Imbalance: 6

T2. Overall Imbalance: 2

T3. Overall Imbalance: 6

T4. Overall Imbalance: 7

T5. Overall Imbalance: 7