Peak Finding In Arrays
2D:

Definition:
Let A be a matrix of size $n \times m$.
Let $A_{i-1,j}$, $A_{i,j}$, $A_{i,j-1}$, $A_{i,j+1}$, and $A_{i+1,j}$, ($i<n, j<m$; $(A_{i-1,j}, A_{i,j}, A_{i,j-1}, A_{i,j+1}, A_{i+1,j}) \in A$) be any five elements in A such that A_i is in the middle, and the other four elements are its neighbors.
An item A_i is a peak iff $A_{i,j} \geqslant A_{i+1,j} \wedge A_{i,j} \geqslant A_{i,j+1} \wedge A_{i,j} \geqslant A_{i,j-1} \wedge A_{i,j} \geqslant A_{i-1,j}$.
Same applies with the elements that are on the edges of the matrix, except they will have less than four neighbors.

Algorithms' asymptotic complexity:
Greedy approach: Worst case, $T(n,m) = \Theta(n \times m)$, since it might need to go through the whole size of the matrix to find the peak.

Recursive approach:
The time it takes to find the largest item in the column is $T(n) = \Theta(n)$, since the algorithm needs to walk through each of the rows at a given column.
=> Time to run the algorithm one time is: $T(n,m) = T(n, m/2) + \Theta(n)$.
As we know from before, the one dimensional peak finding will run logarithmic time.
=> $T(n,m) = \sum_{1}^{\log_2(m)} \Theta(n) = \Theta(n \times \log_2(m))$.

Recursive splitting approach:
Assume, without the loss of generality, that the matrix is of size $n \times n$.
The first time the algorithm searches for the maximum element in the column, it will take $T(n) = \Theta(n)$. Then the recursive step will yield $T(n,n) = T(n, n/2) + \Theta(n)$. The next time it searches for the maximum element in the row, it will take $T(n) = \Theta(n/2)$. After the recursive step it will result in $T(n,n) = T(n/2, n/2) + \Theta(n) + \Theta(n/2)$.
After *i* steps this will result in $T(n,n) = T(n/2^j, n/2^i) + \Theta(n) + \Theta(n/2) + \cdots + \Theta(n/2^{(i+j-1)})$.
We will stop expanding when $\frac{n}{2^i} = \frac{n}{2^j} = 1 \Rightarrow T(\frac{n}{2^i}) = T(\frac{n}{2^j}) = 1$.
¿
=> $T(n,n) = \Theta(n) + \Theta(n/2) + \cdots + \Theta(n/2^{(i+j-1)})$. Since we only care about the highest order term, we are left with the conclusion that $T(n,n) = \Theta(n)$.