

Note: Bonus questions are entirely optional. They can get very challenging and/or fall outside the course curriculum. Interested individuals may attempt.

5. We have encountered this problem in Tutorial 3: Suppose we are given a 2D-array of size m rows by n columns. An element in the array $A[i][j]$ is called a "peak" if it is **greater than or equal to** its adjacent neighbours. You want to find any single peak.

6	8	7	7	1
9	3	1	7	3
8	4	5	3	2

Come up with a (randomized) algorithm which finds a peak. Prove that your algorithm takes less than 350 array accesses with at least 0.5 probability, when $n = m = 100$.

(The "crosshair" algorithm in that week's bonus question takes ≥ 400 accesses.)