Max of Sort of Sorted Arrays

All submitted work must be *done individually* without consulting someone else's solutions in accordance with the University's Academic Dishonesty and Plagiarism policies.

Your genius friend discovered an amazingly fast sorting algorithm using deep quantum neural networks. Unfortunately, due to complicated quantum physics beyond the scope of this unit, the algorithm's output is only sort of sorted. An array S of n integers is sort of sorted if for some x , S[i] is the (x + i mod n)-th smallest number.

(Note that the 0-th smallest number is the smallest number, and the (n-1)-th smallest number is the largest number.)

Your Task

Design a **divide-and-conquer** algorithm that given the array S, outputs the largest number in O(log n) time.

The array S contains **all unique numbers** , and is incorrectly sorted as detailed above.

About the Code

dc_help.py

get_largest_num(S: List[Float]) -> Float

This is the main function, get largest num, which will be the base of your divide and conquer solution to be written.

It takes the array S, which is a list of floating point numbers, and returns the largest number in O(log N) time.

Marking

Your program code will be marked automatically through the Ed submission system.

There are a number of automatic graded unit tests that are run as part of the testing suite.

5%~10% of the marks are assigned purely to correctness. The remaining **90%** is assigned to complexity, where your solution must pass an **O(log N)** complexity threshold. You will be given a mix of public and hidden tests, where public allows you to see the name and outcome of the test, whereas hidden tests only allow you to see the outcome as a pass/fail.