1m 54s left

ALL

1. Math Homework

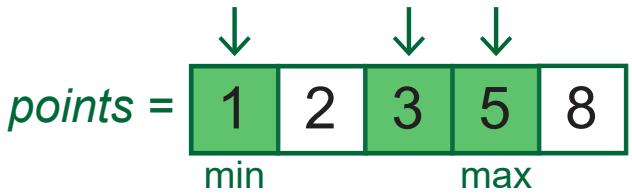
Students have been assigned a series of math problems that have points associated with them. Given a sorted *points* array, minimize the number of problems a student needs to solve based on these criteria.

1. They must always solve the first problem, index *i = 0.*
2. After solving the i^{th} problem, they choose to solve the next problem *(i+1)* or skip ahead and solve the *(i+2)* problem.
3. Students must keep solving problems until the difference between the maximum and minimum points questions solved so far meets or exceeds a specified *threshold*.
4. If students cannot meet or exceed the threshold, they must solve all the problems.

Return the minimum number of problems a student needs to solve.

Example
threshold = 4
points = [1, 2, 3, 5, 8]

If a student solves *points[0] = 1, points[2] = 3 and points[3] = 5,* then the difference between the minimum and the maximum points solved is 5 – 1 = 4. This meets the *threshold,* so the student must solve at least *3* problems. Return *3.*



If the threshold is *7*, again it takes *3* problems solving problems *0, 2* and *4* where *points[4] - points[0] = 8 - 1 = 7*. This meets the *threshold,* so the student must solve at least *3* problems. Return *3.*

If the *threshold* is greater than 7, then there is no way to meet the *threshold.* In that case, all problems need to be solved and the return value is 5.

Function Description
Complete the function *minNum* in the editor.

minNum has the following parameters:
int threshold: the minimum difference required
int points[n]: a sorted array of integers