

774. Minimize Max Distance to Gas Station

Hard 548 77 Add to List Share

You are given an integer array `stations` that represents the positions of the gas stations on the **x-axis**. You are also given an integer `k`.

You should add `k` new gas stations. You can add the stations anywhere on the **x-axis**, and not necessarily on an integer position.

Let `penalty()` be the maximum distance between **adjacent** gas stations after adding the `k` new stations.

Return the **smallest possible value of** `penalty()`. Answers within 10^{-6} of the actual answer will be accepted.

Example 1:

Input: `stations = [1,2,3,4,5,6,7,8,9,10]`, `k = 9`
Output: `0.50000`

Example 2:

Input: `stations = [23,24,36,39,46,56,57,65,84,98]`, `k = 1`
Output: `14.00000`

Constraints:

- $10 \leq stations.length \leq 2000$
- $0 \leq stations[i] \leq 10^8$
- `stations` is sorted in a **strictly increasing order**.
- $1 \leq k \leq 10^5$

Accepted 24,011 Submissions 48,198

Seen this question in a real interview before? Yes No

Companies /

0 ~ 6 months 6 months ~ 1 year 1 year ~ 2 years

Google 2

Related Topics

Array Binary Search

Similar Questions

Koko Eating Bananas Medium

Hide Hint 1

Use a binary search. We'll binary search the monotone function "possible(D) = can we use K or less gas stations to ensure each adjacent distance between gas stations is at most D?"

```
1 * class Solution {
2 *     public double minmaxGasDist(int[] stations, int K) {
3 *         double lo = 0, hi = 1e8;
4 *         while (hi - lo > 1e-6) {
5 *             double mi = (lo + hi) / 2.0;
6 *             if (possible(mi, stations, K))
7 *                 hi = mi;
8 *             else
9 *                 lo = mi;
10 *         }
11 *         return lo;
12 *     }
13 *
14 *     public boolean possible(double D, int[] stations, int K) {
15 *         int used = 0;
16 *         for (int i = 0; i < stations.length - 1; ++i)
17 *             used += (int) ((stations[i+1] - stations[i]) / D);
18 *         return used <= K;
19 *     }
20 * }
```