5463. Best Position for a Service Centre

My Submissions (/contest/weekly-contest-197/problems/best-position-for-a-service-centre/submissions/)

Back to Contest (/contest/weekly-contest-197/)

A delivery company wants to build a new service centre in a new city. The company knows the positions of all the customers in this city on a 2D-Map and wants to build the new centre in a position such that **the sum of the euclidean distances to all customers is minimum**.

Given an array `positions` where `positions[i] = [x_i, y_i]` is the position of the `ith` customer on the map, return *the minimum sum of the euclidean distances* to all customers.

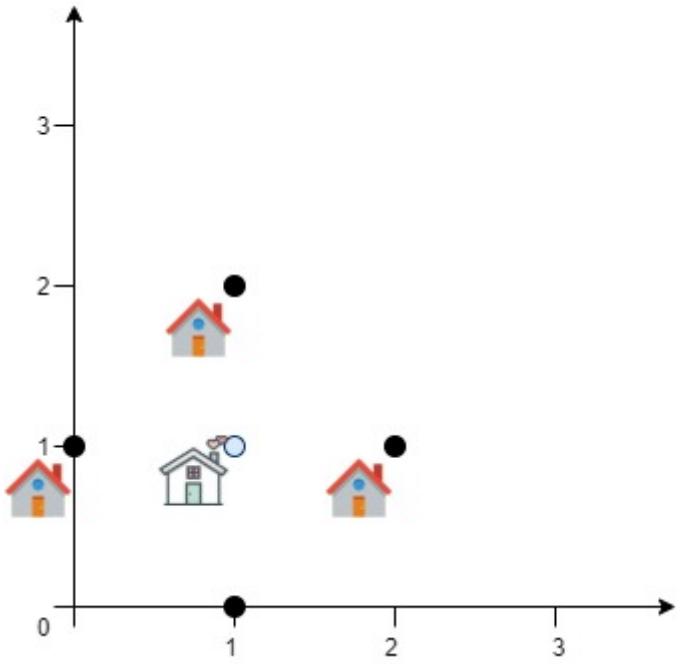
In other words, you need to choose the position of the service centre `[x_centre, y_centre]` such that the following formula is minimized:

$$\sum_{i=0}^{n-1} \sqrt{(x_{centre} - x_i)^2 + (y_{centre} - y_i)^2}$$

Answers within `10^-5` of the actual value will be accepted.

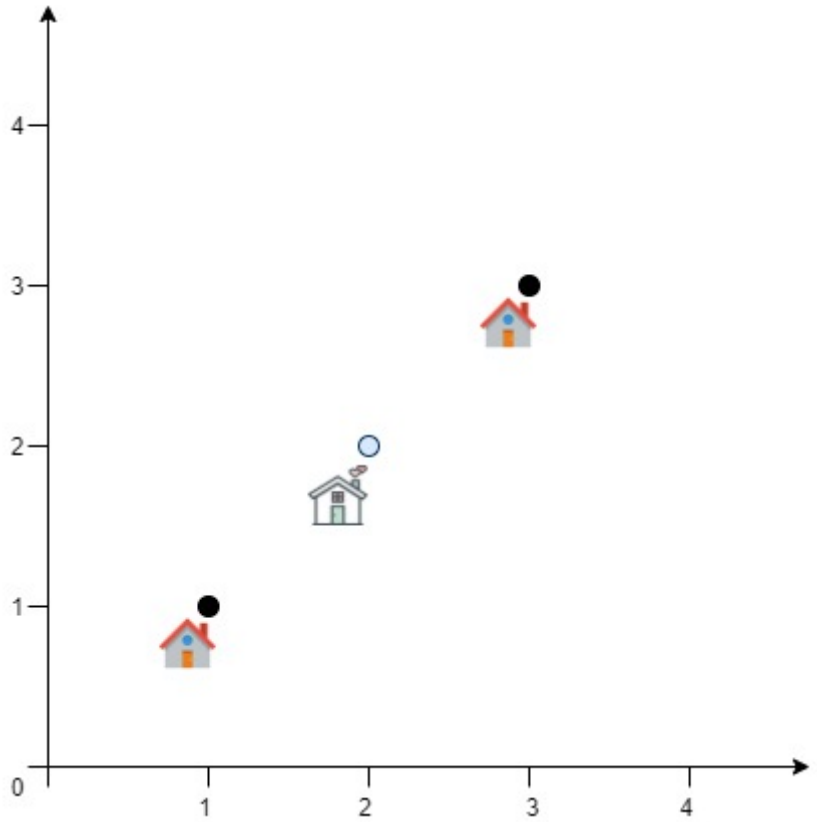
User Accepted:	0
User Tried:	0
Total Accepted:	0
Total Submissions:	0
Difficulty:	Hard

Example 1:



```
Input: positions = [[0,1],[1,0],[1,2],[2,1]]
Output: 4.00000
Explanation: As shown, you can see that choosing [x_centre, y_centre] = [1, 1] will make the di
```

Example 2:




```
Input: positions = [[1,1],[3,3]]
Output: 2.82843
Explanation: The minimum possible sum of distances = sqrt(2) + sqrt(2) = 2.82843
```

Example 3:

```
Input: positions = [[1,1]]
Output: 0.00000
```

Example 4:

```
Input: positions = [[1,1],[0,0],[2,0]]
Output: 2.73205
Explanation: At the first glance, you may think that locating the centre at [1, 0] will ac
Try to locate the centre at [1.0, 0.5773502711] you will see that the sum of distances is
Be careful with the precision!
```

Example 5:

```
Input: positions = [[0,1],[3,2],[4,5],[7,6],[8,9],[11,1],[2,12]]
Output: 32.94036
Explanation: You can use [4.3460852395, 4.9813795505] as the position of the centre.
```

Constraints:

- `1 <= positions.length <= 50`
- `positions[i].length == 2`
- `0 <= positions[i][0], positions[i][1] <= 100`

JavaScript

```
/**
 * @param {number[][]} positions
 * @return {number}
 */
var getMinDistSum = function(positions) {

};
```

Custom Testcase Use Example Testcases

Run Submit