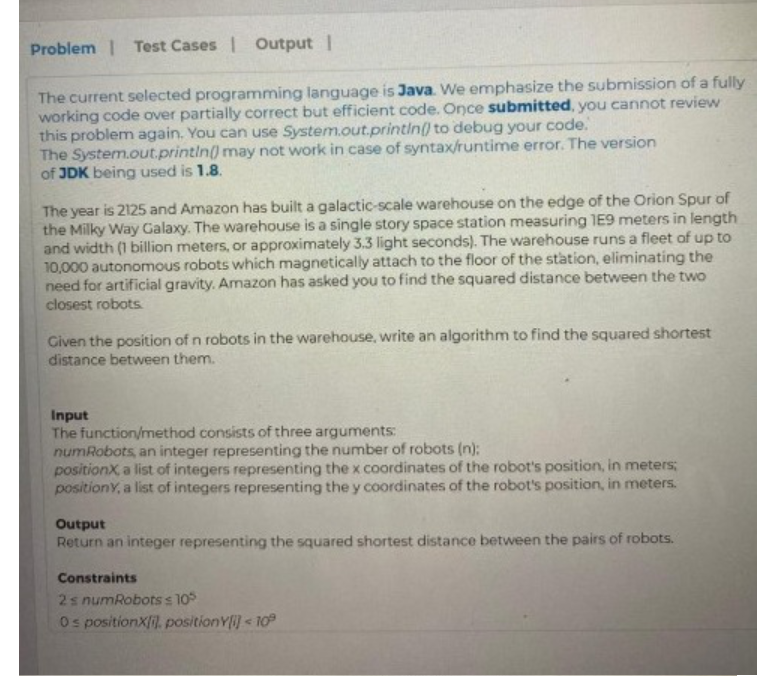
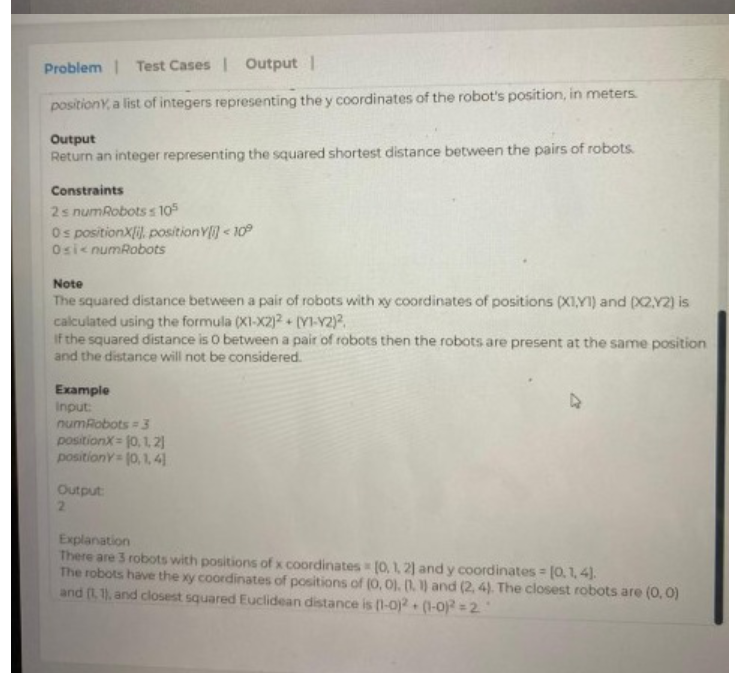
Amazon OA - Shortest mean squared distance between robots Help

LEETCODE : <https://leetcode.com/discuss/interview-question/821708/amazon-oa-shortest-mean-squared-distance-between-robots-help>





Public class Point

{

int x, y;

}

public int closest(int n, int[] positionX, int[] positionY)

{

// Make pairs of points here

Point[] xSorted = new Point[n];

Point[] ySorted = new Point[n];

for (int i = 0; i < n; i++)

{

xSorted[i] = P[i];

ySorted[i] = P[i];

}

Arrays.sort(xSorted, (a,b) -> a.x - b.x);

Arrays.sort(ySorted, (a,b) -> a.y - b.y);

return closestUtil(xSorted, ySorted, n);

}

public int closestUtil(Point Px[], Point Py[], int n) {

if (n <= 3)

return bruteForce(Px, n);

int mid = n/2;

Point midPoint = Px[mid];

// Divide points in y sorted array around the vertical line.

Point[] Pyl = new Point[mid];

Point[] Pyr = new Point[n-mid];

int l = 0, r = 0;

for (int i = 0; i < n; i++) {

if (Py[i].x <= midPoint.x && l < mid)

Pyl[l++] = Py[i];

else

Pyr[r++] = Py[i];

}

// Consider the vertical line passing through the middle point

// calculate the smallest distance dl on left of middle point and dr on right side

int leftdist = closestUtil(Px, Pyl, mid);

int rightdist = closestUtil(Px + mid, Pyr, n-mid);

int mindist = Math.min(leftdist, rightdist);

// Build an array strip[] that contains points close to the line passing through the middle point

Point[] strip = new Point[n];

int j = 0;

for (int i = 0; i < n; i++)

if (Math.abs(Py[i].x - midPoint.x) < mindist)

strip[j] = Py[i], j++;

// Find the closest points in strip. Return the minimum of distance and closest distance is strip[]

return stripClosest(strip, j, mindist);

}

public int bruteForce(Point P[], int n) {

int min = Integer.MAX\_VALUE;

for (int i = 0; i < n; ++i)

for (int j = i+1; j < n; ++j)

if (dist(P[i], P[j]) < min)

min = dist(P[i], P[j]);

return min;

}

public int dist(Point p1, Point p2) {

return Math.abs((p1.x - p2.x)\*(p1.x - p2.x) + (p1.y - p2.y)\*(p1.y - p2.y));

}

public int stripClosest(Point strip[], int n, int mindist) {

for (int i = 0; i < n; i++) {

for (int j = i+1; j < n && (strip[j].y - strip[i].y) < mindist; j++) {

int dist = dist(strip[i],strip[j]);

if (dist < mindist)

mindist = dist;

}

}

return mindist;

}