

#Donut

This algorithm finds the average of the x and y position, which is the (best x, best y) pairing described in the problem statement. The x and y distance from each cop to this coordinate is added to the final result, then printed. The average point from a group will always be the point with the least total distance from every other point, therefore this algorithm will always find the minimum distance. Both the average and the distance is one loop through the positions, the algorithm runs in $O(n)$ time.

Pseudocode

```
int avgX, avgY, totalDist = 0
for each cop in cops:
    avgX += cop.x
    avgY += cop.y
avgX /= cops.length
avgY /= cops.length
for each cop in cops:
    totalDist += | avgX - cop.x | + | avgY - cop.y |
```