

Homework 3 - Problem 1

The algorithm first reads the number of police cars' coordinates on a grid and returns the sum of distances from the median coordinates.

```
main:
  Let size = number of police cars
  Let Set X be a Set holding the x coordinates of the police cars
  Let Set Y be a Set holding the y coordinates of the police cars

  Let xbest = findMedian(X);
  Let ybest = findMedian(Y);

  Let sum = 0
  from index at 1 to size:
    sum = absolute(xbest - X[index]) + absolute(ybest - Y(index))

findMedian:
  Let Set A be a Set parsed in as a parameter
  Let n be the size of A

  If n is odd then
    Return quickSelect(A, n / 2)
  Else
    left = quickSelect(A, n / 2 - 1)
    right = quickSelect(A, n / 2)
    Return (left + right) / 2

quickSelect:
  Let Set A be a Set parsed in as a parameter
  Let n be the size parsed

  If the size of A is 1 then return A[0]

  Let pivot = A[ size of A / 2]
  Let leftCount, rightCount and equalCount = 0

  For every number in A:
    If number < pivot then leftCount++
    Else if number > pivot then rightCount++
    Else equalCount++

  If n < leftCount then:
    Let L be the left subarray containing elements less than the pivot
    quickSelect(A, n)
  Else if n < leftCount + equalCount then:
    Return pivot
  Else
```

```
Let R be the right subarray containing elements less than the pivot  
quickSelect(A, n)
```

```
Return quickSelect(right, n - leftCount - equalCount)
```

The program works because it correctly identifies the medians of police car coordinates, which represent the optimal corner location for the donut store. It accurately calculates the sum of distances to every police car, ensuring a minimum sum, and efficiently handles the task with a time complexity of $O(n)$

Time Estimate: $O(n)$

Explanation:

QuickSelect is the most significant factor in the time complexity, and it is applied twice.

QuickSelect's average-case complexity is $O(n)$, which makes it the dominating factor.

The sum of distances calculation is linear in the number of police car locations, contributing $O(n)$.

QuickSelect: $O(n)$

Sum: $O(n)$

Total: $O(n) + O(n) = O(2n) \Rightarrow O(n)$