

# Analysis: Manhattan Crepe Cart

## Test set 1

Test sets 1 and 2 differ only in how large the available grid area is. In the first test set, people can only be standing in cells with coordinates between 0 and 10, inclusive, in either dimension. Moreover, we can notice that the cart can only be in a cell within this area. The cart's horizontal and vertical coordinates must both be nonnegative per the rules of a problem. Moreover, it cannot have a coordinate larger than 10 (in either dimension). Suppose, for example, that the cart had a horizontal coordinate greater than 10; then that would imply that there must be at least one person standing at horizontal coordinate 10 and facing east. Otherwise, placing the cart at horizontal coordinate 10 would be even better, per the tiebreaker rules. But the rules of the problem do not allow people at (horizontal/vertical) coordinate 10 to face (east/north).

Therefore, to solve test set 1, we can create an array to represent all of the blocks in the allowed area, and initialize each cell's value to 0. Then, for each person, we increment all of the cells of the array that they are walking toward. Finally, we find the cell of the array with the maximum value, using the tiebreaker rules as needed. This solution takes  $O(P \times Q^2)$  time.

## Test set 2: a quadratic solution

In problems that involve multiple dimensions, it is often worth checking whether those dimensions are independent. In this case, they are! A person heading west, for example, gives us a "vote" for the crepe cart being to the west of them, but tells us nothing at all about the north-south location of the cart. So we can solve the two dimensions as separate one-dimensional problems; the horizontal problem includes only the people heading west or east (and their horizontal positions), and the vertical problem includes only the people heading south or north (and their vertical positions). Let us consider the horizontal dimension for now; our arguments also apply to the vertical dimension.

Even if our people are widely spread out along the horizontal axis, the crepe cart can only possibly be in a limited number of horizontal positions. In fact, it must be either at position 0, or at a cell that is one cell to the east of some person. To see why, suppose that the cart is in some other cell  $1 \leq C \leq Q$ . Let  $W$  denote the cell one unit to the west of  $C$ . We know by assumption that  $W$  does not contain a person. But then if we move the cart from  $C$  to  $W$ , we will not be losing any votes (i.e. any person voting for  $C$  is also voting for  $W$ ), and we may even gain votes (if there were people in  $C$  who were heading west). (Notice that a person does not vote for the cell they are in.) Even if we do not gain votes, our tiebreaker gets better. So we should always make this move, and, therefore, we should always move the cart west until it is at 0 or immediately to the east of someone. Observe that this might find only a locally optimal solution, but if we check all such cells, we will surely find the globally optimal solution.

These observations reduce the number of cells we need to check to  $O(P)$  rather than  $O(Q)$ . To check a cell, we can make a linear pass over all of the people, and count whether each one is voting for that cell. One such check takes  $O(P)$  time. Then we choose the cell that got the most votes, breaking a tie if needed by choosing the westernmost of the tied cells. The overall time complexity is  $O(P^2)$  for the one-dimensional problem, and solving it twice (for two dimensions) is still  $O(P^2)$ .

## Test set 2: a (nearly) linear solution

Although the above solution is fast enough to solve test set 2, we can do even better by avoiding making a linear pass over the data for each person.

We can first process our data into a set of tuples like the following: (coordinate, number of people at that coordinate facing west, number of people at that coordinate facing east). Let us denote these as  $(C_i, W_i, E_i)$ . (Remember that for the purposes of the horizontal subproblem, we are ignoring people facing north or south.) Using a hash-based dictionary, this processing takes time linear in  $P$ . As we do this, we should also determine the total numbers  $W$  and  $E$  of people facing west and east.

Then, we sort these tuples in ascending order of their first values. It is probably most convenient to use a common sorting algorithm (or one built into your language), making this step nonlinear. (We will leave a spirited discussion of which sorting algorithms are truly linear for another day.)

Once we have sorted the tuples, we start by considering cell 0 as a candidate location, and we determine the votes for that cell. We know that quantity is equal to  $W$  minus the number of people in cell 0, if any. We make a note of this number of votes and set 0 as our best candidate so far. Then, we look at the first tuple in our sorted list, which represents the people at some cell  $C$  (which might be cell 0). The number of votes for cell  $C + 1$  (the cell one unit to the east of  $C$ ) should be the same as the number of votes we found for cell 0, plus the number of people in cell  $C$  facing east (which is all of them in this case), minus the number of people in cell  $C$  facing west. If cell  $C$  is a better candidate, we store it and its number of votes.

We can do this for every tuple in our list to get our final answer. If we have one or more people on the eastern border, we do not need to check the cell one unit to the east of them, since (as explained at the start of the analysis) the cart can never have a coordinate larger than  $Q$ . Since the check for each entry of the tuple takes constant time, the full pass takes  $O(P)$  time.