

# Analysis: Waffle Choppers

## Test set 1

In test set 1, we are asked to make only one horizontal cut and only one vertical cut. There are  $R - 1$  possible horizontal cuts and  $C - 1$  possible vertical cuts, so there are a total of  $(R - 1) \times (C - 1)$  different ways to make the two cuts. Since  $R$  and  $C$  are both at most 10, there are at most 81 ways, and we can try each one of them separately. For each way, we can count up the chocolate chips in each of the four resulting pieces. If we ever find a way in which that number is the same for all four pieces, the answer is `POSSIBLE`; otherwise, it is `IMPOSSIBLE`.

## Test set 2

In test set 2, we may have to make many horizontal and vertical cuts, and there are too many ways to do this to check. The worst cases occur when  $H$  is close to half of  $(R - 1)$ , and  $V$  is close to half of  $(C - 1)$ ; there could be over  $10^{57}$  ways of making the cuts! So we need another approach.

Let us consider only the horizontal cuts, ignoring the vertical cuts for the moment. These horizontal cuts break the waffle into two or more horizontal "slices". Each of these slices will turn into exactly  $V + 1$  pieces when we make our vertical cuts. Here is the critical observation: if the case is `POSSIBLE`, all of those pieces must have exactly the same number of chips, and since each horizontal slice will produce exactly the same number of pieces, all of those horizontal slices must have exactly the same number of chips. Moreover, we know exactly what that number is; if there are a total of  $C$  chips in the waffle, each of the  $H + 1$  horizontal slices must have exactly  $C / (H + 1)$  chips, or else the case is `IMPOSSIBLE`.

This observation is so powerful that it tells us where we have to make the cuts if the case is `POSSIBLE`! We can make a list of the numbers of chips in each row, then turn that list into a cumulative sum of the total number of chips seen so far. For example, for a waffle with seven rows containing 3, 7, 6, 2, 2, 0, and 10 chips, respectively, the cumulative sum list would be: [3, 10, 16, 18, 20, 20, 30]. In that example, if  $H = 2$ , we know we have to make cuts immediately below rows that bring the total sum to 10 and 20, and we can do so by cutting immediately below the second and fifth rows. (Note that we could instead make our second cut below the sixth row, but this would make no difference.) If  $H = 1$ , though, we need to make our one cut immediately below a row that brings the total sum to 15, and there is no such row. So, after this step, either we will know that the case is `IMPOSSIBLE`, or we will know exactly where to make our horizontal cuts.

Then, we can consider only vertical cuts, using a similar method, and either learn where to make them, or learn that the case is `IMPOSSIBLE`. Even if we know where to make the horizontal and vertical cuts, though, we are not done yet! These cuts might not actually create pieces with the same number of chips. For example, for  $H = 1$ ,  $V = 1$ , and this waffle:

```
. . @@  
. . @@  
@@ . .  
@@ . .
```

we will learn that if the case is `POSSIBLE`, we must make our horizontal cut below the second row, and our vertical cut to the right of the second column. But this creates two pieces with four chips each and two pieces with no chips at all, so the case must be `IMPOSSIBLE`.

To check that each piece has the same number of chips, we can start by turning the list of horizontal cuts into a list of intervals; for example, if we cut below the second and fifth rows in the [3, 10, 16, 18, 20, 20, 30] example from above, then we have the inclusive intervals [1, 2], [3, 5], and [6, 7]. We can do the same for the vertical cuts, and then perform a double iteration over these two sets of intervals, checking each cell within each pair of intervals. We know that each piece must have exactly  $(\text{total \# of chips}) / ((H + 1) \times (V + 1))$  chips if the case is POSSIBLE, so if we ever find a slice in which this is not true, the case is IMPOSSIBLE.

Otherwise, we have finally shown that the case really is POSSIBLE. (Once again, note that even if we had a choice of where to make one or more of our cuts, this must have been due to empty rows or columns, which do not influence the number of chips in each piece.)

(There are other ways to check the pieces; for example, we can make one pass through the data, and, for each cell, calculate the number of chocolate chips in the rectangle that has that cell as its lower right corner, and the upper left corner of the waffle as its upper left corner. Then, to find the number of chips in a certain piece, we can add and subtract the appropriate rectangles from this set.)

In summary, this algorithm involves several steps:

1. Create the row and column sum arrays. We can either make two separate passes through every cell of the waffle, or make one pass and create both arrays at once.
2. Convert the sum arrays into cumulative sum arrays.
3. Check the cumulative sum arrays to find our places to cut.
4. Use the cumulative sum arrays to create interval arrays.
5. Make one pass through every cell of the waffle in a directed way, using the interval arrays, to check whether every piece has the same number of chips.

Steps 1 and 5 above involve looking at each of the  $R \times C$  cells of the waffle, whereas steps 2, 3, and 4 only involve looking at  $R$  or  $C$  values. So steps 1 and 5 dominate the running time, which is  $O(R \times C)$ . (We could not have done better than this anyway, since we clearly have to look at each cell of the waffle at least once to solve the problem.)