Pensieri derivanti dalle discussioni in classe, con punti da chiarire qui e là...

Let k the number of chars that algorithm can store. Let Σ the alphabet. The number of possible states that algorithm can reach are

$$\text{Number of states} = |\Sigma|^k$$

(saving in each cell memory at least one character, so in each cell I can store $|\Sigma|$ different symbol: $|\Sigma|, |\Sigma|, \ldots, |\Sigma|$ k times)

Now, define *family of input* as the set of input (the string that algorithm read) that have the same answer. So, if I show that I can have more than $|\Sigma|^k$ possible family, the algorithm cannot distinguish at least two of them, so the answer is wrong in at least one case (maybe in both if the algorithm works in stupid way).

Try to find a way to generate a family in order to generalize.
Fix k=1. Using the algorithm of exercise 7 (`https://en.wikipedia.org/wiki/Boyer\OT1\textendashMoore_majority_vote_algorithm`) we construct the following string:
`aabbcca`. The algorithm read `a`, store `a - 1` (`1` is the counter). Read `a`, store `a - 2`. Read `bb`, store `a - 0`. Read `cc`, store `c - 2`. Now find `a`, the input ends and the returned value is `c`, that is wrong. We use a specific algorithm but we can prove that every algorithm fails but if we generalize to each k and we show that the possible inputs are greater than possible states, we prove that the algorithm cannot be written.

I try to count the different family but without any success.