Finding the Majority Element

majority element: appears $\geq \frac{N}{2}$ times in stream $e_1, e_2, \ldots, e_N$

Remark: majority element might not exist

↳ algorithm is allowed to return anything in such a case

Boyer-Moore Voting Algorithm

Initialize a counter $c \leftarrow 0$ and stored item $s \leftarrow$ null

For each item e_j in the stream

If $e_j = s$, increment the counter $c \leftarrow c+1$ ← we see current maj element again, counter is incred

Else if $c=0$, $s \leftarrow e_j$ and $c \leftarrow 1$

we need to make sure we only do this if majority element changes

Else, decrement the counter: $c \leftarrow c-1$ ← ok if new element is not majority element

Lemma: If there is a majority element i with frequency $(f_i > \frac{m}{2})$, then the algorithm will return $S=i$.

Idea: For majority element, counter can never be "zeroed out" completely by the other elements (because they are fewer)

How to prove this?

Induction on #elements seen so far?

Claim (?) Having seen n elements, s = majority element among these n elements

we need to add a proposition about this counter c

In the end, it might happen that $c = 1$

11110000

HW: prove it (difficulty: s might not exist, need to be careful with it)