Group assignment 3

David Östling, Mohamed Mahdi, Hamid Noroozi, Filip Döringer Kana

December 11, 2022

Problem description

Consider the following game. Two people, Alice and Bob, each have m (possibly duplicated) integers from the range {1, ..., n}. Let $(a_1, \ldots, a_m)$ denote the numbers that Alice has and $(b_1, \ldots, b_m)$ denote the numbers that Bob has. They want to figure out whether there is a number x that appears strictly more than m times among the $2m$ numbers that they have (i.e., among $a_1, \ldots, a_m, b_1, \ldots, b_m$).

We allow Alice to send a message of $\log^{O(1)}(n+m)$ bits to Bob (so, sending all $a_1, \ldots, a_m$ is not possible). Then, Bob can reply back with a message of $\log^{O(1)}(n+m)$ bits. Then Alice has to either output the number that appears more than m times or determine that such a number does not exist. Describe and analyze a deterministic algorithm for Alice and Bob to do so.

Remarks:

1. You may assume that every player knows m and n in the beginning.
2. We do not care about the running time of players. Thus, they can take as much time to compute their messages as they want. We only care about the size of their messages. So, you do not have to analyze their running time.

Algorithm

The algorithm starts by letting Alice find her most frequent element and storing that element as well as how many times it occurs. We do not care about the running time of this part of the algorithm. After Alice has found these 2 numbers, she can send a message of log(n) + log(m) bits with this information to Bob. This can be done simply by letting the first log(n) bits represent the most occurring number and the final log(m) bits will represent how many times the number occurs. As Bob receives these numbers, he can simply count the number of occurrences of Alice's number he finds in his numbers. If the total number of occurrences is larger than m, he sends back the number he received together with the number of times it occurred in his numbers. If the total number of occurrences is not larger than m, he instead finds the most frequent element of his numbers and sends that together with its count. When Alice receives the message from bob, she counts the number of occurrences of the number Bob sent (which could be the same as the one she sent) and if the count received from Bob added to her own count is larger than m, she outputs that number. If this is not the case, she outputs no majority.

Pseudo-code

Algorithm 1 ALG

Require: $A = (a_1, \dots, a_m)$, $B = (b_1, \dots, b_m)$, m
Ensure: either "no majority" or $x \in A \cup B$ where x has been repeated at least m times

```
Alice : x_a ← Most frequent element in A
Alice : c_a ← Number of occurrences of x_a in A
Alice → Bob : Message containing x_a and c_a
Bob : x_b ← x_a
Bob : c_b ← Number of occurrences of x_a in B
if c_a + c_b ≤ m then
    Bob : x_b ← Most frequent element in B
    Bob : c_b ← Number of occurrences of x_b in B
end if
Bob → Alice : (x_b, c_b)
Alice : c_a ← Number of occurrences of x_b in A
if c_a + c_b ≤ m then
    Alice : return no majority
else
    Alice : return x_b
end if
```

Lemma 1: Message sent is $log^{O(1)}(m + n)$

Proof: We have that we can use any constant exponent, such as 2 for the exponent in the logarithm. We have that the message sent contains at most log(m) + log(n) bits. The smallest value for both m and n is 1. Since log(1) = 0, we have that trivially when either m or n is 1, $log(m) + log(n) < log(m + n)$. For any positive number x, we have that $log(x + \epsilon) > log(x)$ where $\epsilon > 0$. Therefore, we have that

$$log(m) + log(n) < log(m + n) + log(m + n) = 2log(m + n)$$

The second smallest case for the message is when n and m equal 2. For all values where both m and n are greater or equal to 2, we have that

$$2log(m + n) \leq log^2(m + n)$$

When m and n both equal 2, we have that the expression evaluates to $2 * 2 \leq 2^2$. When m and n grow, $log^2(m + n)$ grows much faster than $2log(m + n)$. Thus, using the expressions above, we can conclude that for all values of n and m, we have that $log(m) + log(n) < log^{O(1)}(m + n)$.

Lemma 2: If Alice Returns x; then x has been repeated more than m times in $A \cup B$

Proof: When Alice receives (x_b, c_b) from Bob, she computes c_a as the number of times x_b has occurred in A. So $c_a + c_b$ would be the number of times x_b has occurred in $A \cup B$. The only condition where Alice is returning x, would be when $c_a + c_b > m$, otherwise she is returning "No majority". Therefore, if x is returned, it must have occurred more than m times in $A \cup B$.

Lemma 3: If $\exists x \in (A \cup B)$ such that x has been occurred more than m times; then Alice returns x

Proof: Having x occurred more than m times in $(A \cup B)$ means that $c_{xa} + c_{xb} > m$, where c_{xa} is the number of occurrences of x in A, and c_{xb} is the number of occurrences of x in B.

3.1. x is the most repeated in A

If x is the most repeated in A, then Bob will receive x and c_{xa} from Alice. We have $c_{xa} + c_{xb} > m$, so Bob will send back the same x_a as x_b and it's count in B as c_{xb} to Alice. Again since $c_{xa} + c_{xb} > m$, Alice returns x.

3.2. x is not the most repeated element in A

If x is not the most repeated element in A, it must be the most repeated element in B. Let's assume y is the most repeated in A , then Bob receives y and c_{ya} from Alice, where c_{ya} is the number of occurrences of y in A. If $c_{ya} + c_{yb} > m$, then $x = y$ and the proof follows as 3.1. Otherwise, Bob finds the most frequent element in B which has to be x. So he sends x and c_{xb} to Alice. And since $c_{xa} + c_{xb} > m$, Alice returns x.

Lemma 4: If Alice returns "no majority"; then $\nexists x \in (A \cup B)$ such that x repeats more than m times

Proof: Assume there is a number x occurring more than m times in $(A \cup B)$ with the algorithm outputting no majority. This would mean that x can't be the most frequent number in Alice's list of numbers, since otherwise the algorithm would have found the number and sent it to Bob, who would've counted and found that the combined occurrences are larger than m which he then would've sent to Alice and she would have outputted the number. By the same logic, It cannot be the most frequent element in Bob's list, since he would have found it and sent it to Alice after checking that Alice's number does not occur in total more than m times. Also, since each list contains m elements, it is impossible for any element that is not the most frequent one to occur more than $m/2$ times since that would mean that the most frequent element would also occur more than $m/2$ times. Their combined sum would then be $> m$ which is clearly impossible. Since the sum of two numbers that are both $< m/2$ can never be larger than m, we can conclude that if Alice outputs no majority, then $\nexists x \in (A \cup B)$ such that x repeats more than m times.