Ros Roger, Alexandre

May 26th, 2023

29. We have a data stream that contains all numbers in the set $\{1 \dots n\}$ except for exactly one number. Design an algorithm to determine the missing number. Only one pass through the data stream is allowed, and there is a memory limit of $\mathcal{O}(\log n)$.

Solution:

The trivial and wrong solution would be to store all elements already visited in a bitarray of n cells. This would obviously exceed our memory limitations.

Instead, we are going to initialize a variable we're going to call X and then, for every number i in the data stream, we are going to execute the instruction $X \leftarrow X + i$. Notice that the maximum number of bits needed to encode the sum of all numbers from 1 to n, in binary, is bounded by $\left\lceil \log_2 \left(\frac{n(n+1)}{2} \right) \right\rceil$. Asymptotically, this is $\mathcal{O}(\log_2 \left(n^2 \right)) = \mathcal{O}(2 \log_2 (n)) = \mathcal{O}\left(\log n\right)$. Therefore, the size of X won't exceed the limitations.

After adding up all members of the data stream, we return the value $\frac{n(n+1)}{2} - X$ as the solution to our problem. This is because $\frac{n(n+1)}{2} = \sum_{i=1}^{n} i$, and X is the sum of all $1 \leq i \leq n$ except for exactly one element. The arithmetic operations can all be achieved without exceeding the memory limitations by the same arguments used earlier.