in the following assume $m = n$.
1. use Boyer-Moore voting algorithm, maintain the information (number, count) by segment tree. $O(n \log n)$.
2. the information we maintain forms an (associative) semigroup, so it's mergeable. we need to query range sum on a static array, which needs $O(n\alpha(n))$ by Yao [5]. [1] also gives a $\Theta(n\lambda(k,n))$ ($= O(n\alpha(n))$ for our purpose) time and space algorithm, where $\lambda(k,\cdot)$ is the inverse of a certain function at the $\lfloor \frac{k}{2} \rfloor$-th level of the primitive recursive hierarchy. Then check whether the number we find is valid, by computing the number of occurrence of it in the query interval, using persistent array (or vEB tree) in $O(\log \log n)$. We can also solve this in $O(\frac{\log \log n}{\log \log \log n})$ per query by reducing to the static predecessor problem https://en.wikipedia.org/wiki/Predecessor_problem [2] (but with $O(n^4)$ preprocessing time; in our case the universe $N = n$). In conclusion we get $O(n \log \log n)$.
We can also get $O(n)$ preprocessing and $O(1)$ per query, see my article here: https://zhuanlan.zhihu.com/p/79423299.
3. $O(n \log n)$ preprocessing, $O(1)$ per query [3, 4].

References

[1] Noga Alon and Baruch Schieber. *Optimal preprocessing for answering on-line product queries.* Citeseer, 1987.

[2] Paul Beame and Faith E Fich. Optimal bounds for the predecessor problem. In *STOC*, volume 99, pages 295–304. Citeseer, 1999.

[3] Stephane Durocher, Meng He, J Ian Munro, Patrick K Nicholson, and Matthew Skala. Range majority in constant time and linear space. In *International Colloquium on Automata, Languages, and Programming*, pages 244–255. Springer, 2011.

[4] Gonzalo Navarro and Sharma V Thankachan. Optimal encodings for range majority queries. *Algorithmica*, 74(3):1082–1098, 2016.

[5] Andrew C Yao. Space-time tradeoff for answering range queries. In *Proceedings of the fourteenth annual ACM symposium on Theory of computing*, pages 128–136. ACM, 1982.