1. Hashing. Expected $O(n)$ time.
2. Reduce to integer sorting. After sorting, only need deterministic $O(n)$ time by monotone pointers.
For integer sorting:
$O(n\sqrt{\log\log n})$ randomized [3].
$O(n\log\log n)$ deterministic [2], and $O(n\log\log_w n)$ deterministic [4].

3. For the lower bound, $\Omega(n\log n)$ in the degree-d algebraic decision tree model [1]. We can also reduce the set intersection problem to this, i.e. deciding whether two sets $\{x_1,\ldots,x_n\}$ and $\{y_1,\ldots,y_n\}$ intersect, by negating one of the sets.

References

[1] AlexanderC. L. Chan, William I. Gasarch, and Clyde P. Kruskal. Refined upper and lower bounds for 2-sum. 2004.

[2] Yijie Han. Deterministic sorting in $O(n\log\log n)$ time and linear space. In *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*, pages 602–608. ACM, 2002.

[3] Yijie Han and Mikkel Thorup. Integer sorting in $O(n\sqrt{\log\log n})$ expected time and linear space. In *The 43rd Annual IEEE Symposium on Foundations of Computer Science, 2002. Proceedings.*, pages 135–144. IEEE, 2002.

[4] Mikkel Thorup, Or Zamir, and Uri Zwick. Dynamic ordered sets with approximate queries, approximate heaps and soft heaps. In *46th International Colloquium on Automata, Languages, and Programming (ICALP 2019)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2019.