1. Consider the following problem in the streaming model.
Median: Given a stream $S = \langle a_1,\ldots,a_m \rangle$ of m distinct items over the universe [n], compute a median of S.
Prove that for $m < n/2$ any deterministic streaming algorithm that solves Median exactly must use $\Omega(m \log(n/m))$ bits in the worst case.

2. Consider the following problem in the streaming model.
Element Uniqueness: Given a stream $S = \langle a_1,\ldots,a_m \rangle$ over the universe [n], with $m <= n$, decide if all items in S are distinct.
Either prove that any deterministic streaming algorithm that solves Element Uniqueness exactly must use $\Omega(m \log(2n/m))$ bits in the worst case, or give a deterministic streaming algorithm that solves Element Uniqueness exactly using a sub-linear number of bits. If you give an algorithm, you should also prove its correctness and analyze the number of bits of storage it uses.

3. Consider the following problem in the streaming model.
Two Missing Items: Given a stream $S = \langle a_1,\ldots,a_{n-2} \rangle$ over the universe [n] in which all items in S are different, compute the items j_1, j_2 in [n] that are missing from S.
Note that only streams of length n - 2 are considered and that all items in the stream are distinct, which implies there are exactly two missing items.
Either prove that any deterministic streaming algorithm that solves Two Missing Items exactly must use $\Omega(n)$ (n) bits in the worst case, or give a deterministic streaming algorithm that solves Two Missing Items exactly using a sub-linear number of bits. If you give an algorithm, you should also prove its correctness and analyze the number of bits of storage it uses.

4. Present a parallel algorithm for removing the duplicate items that appear in a sequence. The input to the algorithm is a sequence, and the output is a new sequence containing exactly one copy of every item that appears in the input sequence. It is assumed that the order of the items in the output sequence does not matter.

5. Consider the problem of labeling the connected components of an undirected graph. The problem is to label all the vertices in a graph G such that two vertices u and v have the same label if and only if there is a path between the two vertices. Propose a parallel algorithm for this problem.

6. Discrete Fourier Transform (DFT) has a long history of parallel algorithms. Propose a method to parallelize the Fast Fourier Transform (FFT) algorithm for solving the DFT.

7. Suppose we have joined an online shopping site, and that there are n cars that we are rather interested in. We would like to buy the best one. (We are assuming that cars are comparable, and that there is a consistent way, after checking two cars, to decide which one is better.) We could rent all of them, one after the other, for one day and then pick the best, but the rule is that we cannot rent simultaneously and after one day of rent either we choose the car or ignore it forever! How can we maximize the probability of ending up buying the best car? Propose a competitive algorithm and analyze its ratio.