# $12^{\text{TH}}$ TUTORIAL ON RANDOMIZED ALGORITHMS

Yao's minimax principle and its application to lower bounds

**1**. *Searching.* Show a lower bound on the expected running time of any Las Vegas comparison-based algorithm for searching in a sorted array.

**2**. *Sorting.* Show a lower bound on the expected running time of any Las Vegas comparison-based algorithm for sorting $n$ numbers.

**3**. *Three consecutive 1s.* Given a binary string of length $n$, the goal is to determine whether or not there are three consecutive 1s. Show a lower bound on the expected number of steps, where in one step, the algorithm can examine one bit.

**4**. *Majority element in a query model.* Given a list of values $v_1, \ldots, v_n$, the goal is to find an index $i$, if one exists, such that the value $v_i$ occurs more than $n/2$ times in the list. Determine a lower bound on the expected running time of any Las Vegas algorithm that solves the problem, but is restricted to only ask equality queries; that is, in each step the algorithm specifies indexes $i, j$ and is told whether $v_i = v_j$ or not. The algorithm cannot access $v_i$'s in any other way.

**5**. *Perfect matching.* Let $G$ be an $n$-vertex graph for an even $n$ such that we only have a query access to the edges, namely, we can only ask whether two nodes are connected or not. Show that $\Omega(n^2)$ queries are needed in expectation for any algorithm that correctly determines whether $G$ has a perfect matching or not.