Problem 1

Consider the problem of sorting an array $A = (a_1, a_2, \ldots, a_n)$ of elements with multiplicities $(m_1, m_2, \ldots, m_k)$. That is, $A$ is made up of $k$ distinct elements $(y_1, y_2, \ldots, y_k)$, where $y_i$ occurs $m_i$ times in $A$. Prove that any algorithm in the comparison model requires $\Omega(n \log n - \sum_{i=1}^{k} m_i \log m_i)$ comparisons to sort $A$.

Problem 2

Given a set of $k$ sorted arrays $\{A_1, \ldots, A_k\}$, where the combination of the $k$ arrays has $n$ elements, give an $O(n \log k)$ algorithm that produces a single sorted array containing all $n$ elements. Hint: use a priority queue.

Problem 3

Perform QuickSelect to find the 3rd smallest element in the array $A = [8, 17, 10, 1, 6, 20, 2, 9, 7, 13]$. 