CS 240: Data Structures and Data Management **Spring 2026**

Tutorial 04: June 12

1. **Multiplicity sorting** Consider the problem of sorting an array A of n elements with multiplicity n/k. That is, A consists of k distinct elements $(y_1, y_2, \ldots, y_k)$, where each y_i occurs n/k times in A. Prove that any algorithm in the comparison model requires $\Omega(n \log k)$ comparisons to sort A in the worst-case.
   Note: $\forall m \geq 0, \left(\frac{m}{e}\right)^m \leq m! \leq m^m$.

2. **Priority queue lower bounds** Does there exist a priority queue for which the `insert` and `delete-max` operations require only $O(\log \log n)$ operations in the worst case?

3. **Almost sorted arrays** Consider the problem of sorting an array A of n distinct elements, each of which is at most k positions away from its location in the sorted order. That is, if j is the index of $A[i]$ in the sorted order of A, then $-k \leq j - i \leq k$.

   (a) Design an algorithm to sort A in time $O(n \log k)$.

   (b) Prove that any algorithm in the comparison model requires $\Omega(n \log k)$ comparisons to sort A in the worst case.