Tutorial 03 - Sorting & Average Case
CS 240E Winter 2022
University of Waterloo
Monday, January 24, 2022

1. **Partially Sorted:**
   Let $0 < \epsilon < 1$. Suppose that we have an array A of n items such that the first $n - n^{\epsilon}$ items are sorted. Describe an $O(n)$ time algorithm to sort A.

2. **String Comparison:**

   Let A and B be two bitstrings of length n (modelled here as arrays where each entry is 0 or 1). A *string-compare* tests whether A is smaller, larger, or the same as B and works as follows:

```
Algorithm 1: str-cmp(A, B, n)
for ($i = 0; i < n; i++$) do
    if ($A[i] < B[i]$ then return "A is smaller"
    if ($A[i] > B[i]$ then return "A is bigger"
return "They are equal"
```

   Show that the average-case run-time of *str-cmp* is in $O(1)$. You may use without proof that $\sum_{i \geq 0} \frac{i}{2^i} \in O(1)$.

3. **Fraction Select:**
   Let L denote a sorted array of n distinct integers that are pairwise coprime. Given L and an integer k between 1 and $\frac{n(n-1)}{2}$, write a function that produces a pair (i, j), with $i < j$, such that $\frac{L[i]}{L[j]}$ is the k-th smallest fraction that can be made from elements in L. The algorithm should run in $O(k \log k)$ time.