SSN College of Engineering

Department of Computer Science and Engineering

CS1403 — Design and Analysis of Algorithms

2019 – 2020

Session — 03

January 8, 2020

- This homework is due by 4pm on January 08, 2020
- Grace period may be given up to midnight of January 08, 2020
- You can upload only one ZIP file
- The naming convention is "<Your first name (first letter capital and all the other letters small)>-CS1403-S03.zip"

---

1. In several applications there is a need to compare two different rankings. For example, let us say every visitor to Chennai Book Fair is asked to rank a set of books. How do we quantify the similarity (or dissimilarity) between the rankings given by two visitors v_1 and v_2? If the rankings are exactly the same, then the dissimilarity measure should be low. When they are exactly the opposite (when one ranking is the reverse of the other), the dissimilarity must be high. In other cases, the measure should be a smooth interpolation between these two extremes.

   One method of quantifying the dissimilarity is as follows. Consider the ranking labels for the books as per the ranking given by v_1. For example, if a book B_i is ranked fifth as per v_1, then the label of B_i is 5. Now consider the sequence of labels of the books as per the rankings of v_2. Let us take a specific case where the sequence is $\langle 3, 1, 2, 5, 4\rangle$. This means that the book ranked as first by v_1 is ranked as second by v_2. The book that received the third rank in the rankings of v_1 is ranked as first by v_2 and so on. Now, dissimilarity can be measured as the number of pairs that are out of order. In our example, the dissimilarity measure is 3, since there are three pairs $(3, 1), (3, 2), (5, 4)$ that are out of order. The number of out-of-the-order pairs provide a measure that smoothly interpolates between complete agreement (0 pairs) and complete disagreement (${}^nC_2$ pairs).

   So, here is the concrete problem. Given a sequence of n distinct numbers, count the number of out-of-the-order pairs in that sequence.

   Some examples are,

$$\langle 1, 2, 3, 4, 5\rangle \mapsto 0$$
$$\langle 2, 4, 1, 3, 5\rangle \mapsto 3$$
$$\langle 3, 1, 2, 5, 4\rangle \mapsto 3$$
$$\langle 4, 5, 2, 1, 3\rangle \mapsto 7$$
$$\langle 5, 4, 3, 2, 1\rangle \mapsto 10$$

(a) Design and implement (in Python) an algorithm using brute-force examination of all possible pairs of labels

(b) Analyse your algorithm and report the run-time complexity in asymptotic notation

(c) (OPTIONAL) Perform empirical analysis to verify the run-time complexity

(d) Design and Implement (in Python) an improved version (run-time complexity should be less) using any design technique what we have discussed so far in the class

(e) Analyse your improved algorithm and report the run-time complexity in asymptotic notation

(f) (OPTIONAL) Perform empirical analysis to verify the run-time complexity