Homework #2
Introduction to Algorithms/Algorithms 1
600.363/463
Spring 2016

Due on: Thursday, February 11th, 11.59pm
Late submissions: will NOT be accepted
Format: Please start each problem on a new page.
Where to submit: On blackboard, under student assessment
Please type your answers; handwritten assignments will not be accepted.
To get full credit, your answers must be explained clearly,
with enough details and rigorous proofs.

February 4, 2016

1 Problem 1 (10 points)

Given two unsorted integer arrays, A and B, of size n, where A has no repeated elements and B has no repeated elements, give an algorithm that finds k-th smallest entry of their intersection, $A \cap B$. For full credit, you need to provide an algorithm that runs in $O(n \log n)$ time with correctness proof and running time analysis.

2 Problem 2 (15 points)

Given two sorted integer arrays, A and B, of size n, give an efficient algorithm that finds k-th smallest entry of their union, $A \cup B$. For full credit, you need to provide an algorithm that runs in $O(\log n)$ time with correctness proof and running time analysis.

3 Problem 3 (10 points)

You are given one unsorted integer array $A = \{a_i\}_{i=1}^{n}$ of size n. Provide an algorithm that finds

$$r = \max_{1 \le i,j \le n} |a_i - a_j|$$

using at most $O(n)$ comparisons on the worst case input (5 points) or an algorithm which uses at most $\frac{3}{2}n$ comparisons on the worst case input (10 points). Correctness proof and running time analysis are required in both cases.

4 Problem 4 (15 points)

You are given one unsorted integer array A of size n. You know that A is almost sorted, that is it contains at most m inversions, where inversion is a pair of indices (i, j) such that $i < j$ and $A[i] > A[j]$.

1. To sort array A you applied algorithm Insertion Sort. Prove that it will take at most $O(n + m)$ steps.

2. What is a maximum possible number of inversions in the integer array of size n?