Homework #4
Introduction to Algorithms/Algorithms 1
600.363/463
Spring 2016

Due on: Thursday, Feb 25th, 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 19, 2016

1 Problem 1 (12 points)

You are given $k = O(1)$ unsorted integer arrays $A_1, A_2, \ldots, A_k$. Each array A_i contains n elements s.t. $A_i[j] \in \{1 \ldots n\}$. Another array C is defined as

$$C[j] = \prod_{i=1}^{k} A_i[j] = A_1[j] * A_2[j] * \ldots * A_k[j].$$

Write an algorithm that sorts array C in $O(n)$ time. Prove correctness and provide running time analysis.
For example, consider the case when $k = 3$ and $n = 4$. $A_1 = \{1, 1, 3, 2\}$, $A_2 = \{1, 1, 2, 1\}$ and $A_3 = \{3, 2, 4, 1\}$, then $C = \{3, 2, 24, 2\}$, and we want to sort C in $O(n)$ time.

2 Problem 2 (13 points)

You are given an array A of n integer. All numbers except for $O(\log n)$ are in the range $[1, 1000n^2 - n]$. Find an algorithm that sorts an array A in $O(n)$ time in the worst case. Provide running time analysis and correctness proof for your algorithm.

3 Problem 3 (13 points)

Given an array A of n integers from the range $[1, m^3]$. A is stored as m pairs $(\text{item}, \# \text{ of instances})$. For example, if initially array was stored as

$$1, 2, 2, 3, 2, 3, 2, 1, 2, 5, 1, 5, 4, 3, 2, 1, 7, 2, 3, 6$$

then its new representation is:

$$(1, 4), (7, 1), (2, 7), (5, 2), (4, 1), (3, 4), (6, 1)$$

which you can read as item 1 appears 4 times in A, item 7 appears only once in A, item 2 appears 7 times in A, and so on. Provide an algorithm that finds k-th smallest integer in the array in $O(m)$ time with running time analysis and correctness proof for your algorithm.
Note, there is no dependency on n in time complexity.

4 Problem 4 (12 points)

Given two integer arrays A of size n and B of size k, and knowing that all items in the array B are unique, provide the algorithm which finds indices $j' < j''$, such that all elements of B belong to $A[j' : j'']$ and value $|j'' - j'|$ is minimized or returns zero if there is no such indices at all.
For example, consider array $A = \{1, 2, 9, 6, 7, 8, 1, 0, 0, 6\}$ and $B\,\{1, 8, 6\}$, then you can see that $B \subseteq A[1 : 6]$ and $B \subseteq A[4 : 7]$, but at the same time $7 - 4 < 6 - 1$, thus algorithm should output $j' = 4$ and $j'' = 7$.
For full credit, your algorithm must run in $O(nk)$ and use at most $O(n)$ of extra memory. Prove correctness and provide running time analysis.