College of Computer & Information Science — Fall 2014
Northeastern University — Handout 27
CS7800: Advanced Algorithms — 8 December 2014

Final

Due by 5:10 PM, Wednesday, 10 December 2014

- **Honor code:** This exam is open-notes and open-book. However, *searching for solutions or ideas online, and collaboration of any kind are strictly prohibited.* ("Collaboration" includes, for example, discussion or exchange of material related to the problems on the exam with anyone other than the instructor.) If you have any questions or clarifications, please ask me.

  Please print a copy of this page and sign below indicating that you are strictly abiding by the honor code. Submit the signed page to me as a hard copy or by email as a scanned copy by the deadline.

  Name: ________________________________

  Signature: ________________________________

- **Policy on cheating:** Students who violate the above rules on scholastic honesty are subject to disciplinary penalties. Any student caught cheating will receive an **F** (failing grade) for the course, and the case will be forwarded to the Office of Student Conduct and Conflict Resolution.

- **Writing of the solutions:** Both hand-written and typeset submissions will be accepted. If hand-written, please make sure that your writing is legible and neat. Thanks!

- **Presentation of solutions:** While describing an algorithm, you may use any of the algorithms covered in class or in the text as a subroutine, without elaboration.

- **A note on grading:** Your grade on any problem that asks you to design an algorithm will be determined on the basis of its correctness, clarity of the description, and how well it satisfies the desired properties. In case you are not able to present an algorithm that has the desired properties, give the best algorithm you have designed. Show your work, as partial credit may be given.

- **Parts, problems and points:** This exam has four problems, worth a total of 30 points, and 4 bonus points. This exam counts for 30% of the total grade.

- **Submission:** Please submit by giving me a hard-copy, leaving a hard-copy in my office mailbox, or emailing me an electronic version.

- **Good Luck!**

Problem 1. (5 points + 4 bonus points) Finishing up sorting

We are given an array A with n distinct integers, which is partially sorted: each element in A is within distance m of its position in the sorted order (increasing order). Formally, the input array A satisfies the following condition: For $i \in [1, n]$, the rank of $A[i]$ is in the range $[\max\{1, i - m\}, \min\{n, i + m\}]$.

(a) **(5 points)** Suppose we know m. Design an $O(n \lg m)$ time algorithm to solve the problem. Briefly justify the worst-case running time of your algorithm. You need not prove the correctness of your algorithm.

For partial credit, you may give an algorithm that has a worse running time.

(b) **(2 bonus points)** Now suppose we *do not know* m. Using the algorithm of part (a) as a black box, design an algorithm that takes $O(n \lg^2 m)$ time to sort the array. Analyze the worst-case running time of your algorithm. You need not prove the correctness of your algorithm.

For **2 additional bonus points**, give an algorithm that has an asymptotically better running time than $O(n \lg^2 m)$.

Problem 2. (10 points) Warcraft reimagined

As a programmer for the *World of Warcraft* game, you are designing a battle scene in which a team of m Humans fights against a team of invading $n \le m$ Orcs. Your goal is to assign, for each Orc, a distinct Human to battle that Orc; i.e., you will be selecting exactly n of the m Humans to fight the Orcs. For $1 \le i \le m$, you know the *power* p_i of the ith Human, and for $1 \le j \le n$, you know the power q_j of the jth Orc. Assume that all the p_is are distinct integers, and all the q_js are distinct integers.

To make for the most interesting battle scene, you would like to assign Humans to battle Orcs so that the sum, over all the n assigned Human-Orc pairs, of the *absolute differences of their powers is minimized.*

For example, if there are 4 Humans and 2 Orcs, and Human 2 is assigned to Orc 1 and Human 3 is assigned to Orc 2, then the sum of the absolute differences of their powers equals $|p_2 - q_1| + |p_3 - q_2|$.

(a) **(3 points)** Show that in any optimal solution, for any two Orcs i and j, if Orc i is assigned Human k, Orc j is assigned Human ℓ, and $q_i < q_j$, then $p_k < p_\ell$.

(b) **(7 points)** Using part (a), design a polynomial-time algorithm to solve the problem. You need not prove the correctness of the algorithm. You need not analyze its running time. (*Hint:* Use dynamic programming.)

Problem 3. (10 points) Resource allocation

Here is a class of problems that often arises in the allocation of resources. You would like to complete a collection P of projects. Each project requires certain specific resources, drawn from a set R. Let S_i denote the set of resources needed by the ith project. Each resource can be allocated to at most one project.

The goal of the *resource allocation problem* is to determine if there exist k projects such that each project can be allocated the resources it needs under the constraint that each resource is allocated to at most one project. Consider the resource allocation problem in the following two special cases.

(a) **(5 points)** Each resource is required by at most two projects. That is, for each resource r, there exist at most two projects i and j such that $r \in S_i$ and $r \in S_j$. Show that this special case of the problem is NP-complete.

(b) **(5 points)** There are two types of resources; that is, the set R is the disjoint union of the sets R_1 and R_2, where R_1 is the set of all resources of the first type, and R_2 is the set of all resources of the second type. Further, suppose that each project requires at most one specific resource from R_1 and at most one specific resource from R_2; so for each project i, $|S_i \cap R_1| \leq 1$ and $|S_i \cap R_2| \leq 1$.

Give a polynomial-time algorithm for solving this special case. You need not prove the correctness of the algorithm. You need not analyze its running time.

Problem 4. (5 points) Finding a fitting line

Design a polynomial-time algorithm for the following problem. Given a set of n points (a_i, b_i), $1 \leq i \leq n$ find a line $\alpha x + y = \beta$ (i.e., find α and β) that best fits the n points in the following sense: the line minimizes

$$\max_i |\alpha a_i + b_i - \beta|.$$

Assume that α_i's and β_i's are all positive integers.

You need not prove the correctness of your algorithm. You need not analyze the running time of your algorithm.