Quiz 2

This take-home quiz contains 5 problems worth 25 points each, for a total of 125 points. Your exam is due between 10:00 and 11:00 A.M. on Monday, November 22, 2004, in 32-123. Late exams will not be accepted unless you obtain a Dean’s Excuse or make prior arrangements with your recitation instructor. You must hand in your own exam in person.

The quiz should take you about 12 hours to do, but you have five days in which to do it. Plan your time wisely. Do not overwork, and get enough sleep. Ample partial credit will be given for good solutions, especially if they are well written. Of course, the better your asymptotic running-time bounds, the higher your score. Bonus points will be given for exceptionally efficient or elegant solutions.

Write-ups: Each problem should be answered on a separate sheet (or sheets) of 3-hole punched paper. Mark the top of each problem with your name, 6.046J/18.410J, the problem number, your recitation time, and your TA. Your solution to a problem should start with a topic paragraph that provides an executive summary of your solution. This executive summary should describe the problem you are solving, the techniques you use to solve it, any important assumptions you make, and the running time your algorithm achieves.

Write up your solutions cleanly and concisely to maximize the chance that we understand them. Be explicit about running time and algorithms. For example, don’t just say you sort n numbers, state that you are using heapsort, which sorts the n numbers in $O(n \log n)$ time in the worst case. When describing an algorithm, give an English description of the main idea of the algorithm. Use pseudocode only if necessary to clarify your solution. Give examples, and draw figures. Provide succinct and convincing arguments for the correctness of your solutions. Do not regurgitate material presented in class. Cite algorithms and theorems from CLRS, lecture, and recitation to simplify your solutions.

Part of the goal of this exam is to test engineering common sense. If you find that a question is unclear or ambiguous, make reasonable assumptions in order to solve the problem, and state clearly in your write-up what assumptions you have made. Be careful what you assume, however, because you will receive little credit if you make a strong assumption that renders a problem trivial.

Bugs, etc.: If you think that you’ve found a bug, send email to 6046-staff@theory.csail.mit.edu. Corrections and clarifications will be sent to the class via email. Check your email daily to avoid missing potentially important announcements. If you did not receive an email last night reminding you about Quiz 2, then you are not on the class email list, please let your recitation instructor know immediately.

Policy on academic honesty: This quiz is “limited open book.” You may use your course notes, the CLRS textbook, lecture videos, basic reference materials such as dictionaries, and any of the handouts posted on the course web page, but no other sources whatsoever may be consulted. For
example, you may not use notes or solutions from other times that this course or other related courses have been taught, or materials on the World-Wide Web. (These materials will not help you, but you may not use them anyhow.) Of prime importance, you may not communicate with any person except members of the 6.046 staff about any aspect of the exam until after noon on Monday, November 22, even if you have already handed in your exam.

If at any time you feel that you may have violated this policy, it is imperative that you contact the course staff immediately. It will be much the worse for you if third parties divulge your indiscretion. If you have any questions about what resources may or may not be used during the quiz, send email to 6046-staff@theory.csail.mit.edu.

PLEASE REREAD THESE INSTRUCTIONS ONCE A DAY DURING THE EXAM.
GOOD LUCK, AND HAVE FUN!
Problem 1. A Dynamic Set

A dynamic set \( Q \) contains items, where each item \( x \) has an associated key \( key[x] \). The dynamic set \( Q \) supports the following operations:

- **INSERT** \( (x, Q) \): Insert item \( x \) into \( Q \).
- \( x \leftarrow \text{EXTRACT-OLDEST}(Q) \): Remove and return the oldest item \( x \in Q \). (The oldest item is the one inserted least recently.)
- \( x \leftarrow \text{FIND-MAX}(Q) \): Return (but do not remove) the item \( x \in Q \) for which \( key[x] \) is maximal.

Design a data structure for \( Q \) that can perform any sequence of \( n \) operations efficiently.

Problem 2. Great Minds Need Coffee

It is a beautiful autumn morning, and Professor Indyk has decided to walk from his apartment to MIT where he will give a lecture for 6.046. To prepare for this trip, the professor has gone online and downloaded a map of the \( n \) roads and \( m \) intersections in the Boston area. For each road \( e \) connecting two intersections, the map lists the length \( w[e] \) of the road in meters. (Recall that in the Boston area, an arbitrarily large number of roads can meet at a single intersection, e.g., Davis Square.)

Like many great theorists, Professor Indyk cannot walk more than 1000 meters without sitting down and assuaging his caffeine addiction. (This is especially true before 9:30 A.M.) Fortunately, the Moonbucks Coffee Company has recently opened a large number of coffee shops throughout the Boston area. Let \( B = \{b_1, b_2, \ldots, b_k\} \) be the set of \( k \) intersections hosting Moonbucks coffee shops.

Thus, Professor Indyk is looking for a route from his apartment (located at one given intersection) to the Stata Center (located at another given intersection) in which he never travels more than 1000 meters without passing through an intersection with a coffee shop. Help the professor get to lecture on time by designing an efficient algorithm to find the shortest acceptable route from his apartment to the Stata Center.

Problem 3. Radio 107.9 FM

The Eccentric Motors (EM) corporation is about to roll out their 2005 year car model when they discover that many of their automobiles have faulty radio tuners that cannot access the highest frequencies in the FM spectrum. In particular, these defective radios cannot receive radio stations broadcast at 107.9 FM. Replacing the defective radios would delay the roll out of the 2005 model, which would cost the company millions of dollars in lost sales. Fortunately, not every city has a 107.9 FM station. Therefore, EM has decided to send the defective cars only to those dealerships that are out of range of all 107.9 FM stations.

Eccentric Motors has given you a list of their \( n \) dealerships and has asked you to locate the dealerships that are out of range of all 107.9 FM stations. The list is organized so that dealership \( i \) is \( D_x[i] \) miles east and \( D_y[i] \) miles north of St. Louis, Missouri. (Negative values of \( D_x[i] \) and \( D_y[i] \) denote miles west and south, respectively.) You have contacted the FCC, which has given you the
location of the \( k \) radio towers that broadcast at 107.9 FM. Tower \( j \) is located \( T_x[j] \) miles east and \( T_y[j] \) miles north of St. Louis, Missouri, and it broadcasts at a signal strength that allows the signal to be received within a radius of \( T_r[j] \) miles from the tower. FCC regulations guarantee that no point is able to receive signals from two different 107.9 FM broadcast towers.

Design an efficient algorithm to locate the EM dealerships that are out of range of all 107.9 FM broadcast towers.

**Problem 4. Isolating Kryptonite**

Professor Luthor has obtained a meteorite fragment that contains high levels of a useful Kryptonite isotope. The fragment is a perfect cube \( n \) centimeters on each side. The professor would like to distribute this fragment to \( m \) secret laboratories for study in the hopes that the Kryptonite isotope can be synthesized in large quantities. The Kryptonite is not evenly distributed throughout the meteorite, however. The density of the Kryptonite at coordinate \((i, j, k)\) is given by \( D(i, j, k) \), that is, \( D(i, j, k) \) gives the amount of isotope in a cubic centimeter with origin \((i, j, k)\).

Professor Luthor has obtained access to a machine that will cut across the meteorite fragment at a given position, thereby dividing it into exactly two pieces. The machine only cuts at right angles (parallel to the \(-x-, y-, \text{ or } z\)-axis) and on centimeter boundaries. Thus, each cut results in two rectangular parallelepipeds with integral dimensions.

Design an efficient algorithm for dividing the meteorite into \( m \) pieces so as to maximize the minimum amount of Kryptonite contained in any piece. (Partial credit will be given for solving the analogous two-dimensional problem.)

*Clarification:* The machine can only be used to cut a single piece into two smaller pieces. That is, after cutting the meteorite into two pieces, you must separate the pieces; you can then perform different cuts on each piece. Also, you cannot glue pieces back together.
Problem 5. Finding Coolmail Users

Macrohard Corporation has decided to start a free email service known as Coolmail. Coolmail users choose their favorite $k$-digit number $x$ as their user ID and get the email address $x@coolmail.com$. For instance, with $k = 9$ a user might chose 314159265 as her user ID and be given the address 314159265@coolmail.com. Since many people find it hard to remember $k$-digit numbers, Coolmail provides the following helpful service. Whenever an email is sent to an address $y@coolmail.com$, where $y$ is not a valid user ID, Coolmail finds a nearest valid user ID $x$, that is, an $x$ that minimizes $|x - y|$, and sends the reply, “Sorry, but $y@coolmail.com$ is not a valid address. Did you mean to type $x@coolmail.com$?”

It has come to the attention of the opportunistic Professor Ralsky that many Coolmail users are paying too much interest on their mortgages. He would therefore like to send every Coolmail user a message informing him or her that “You can refinance your mortgage at the amazing rate of only 3.49%!!” Unfortunately, the professor does not know the email addresses of all Coolmail users. Let $0 < x_1 < \cdots < x_n < 10^k$ be the user ID’s of the $n$ Coolmail users, where $n$ is unknown to the professor. He has hired you as an MIT student intern to design an efficient algorithm to compute, with the help of Coolmail’s automated reply service, the email addresses of all the Coolmail users. Minimize the worst-case number of emails required by your algorithm.