

Practice Quiz 2 for Spring 2012

These problems are four of the five problems from the take-home exam given in spring 2011, along with the full instructions given with the exam so that you have a sense of what the take-home exam will be like.

Because of the order in which we're doing things this term, you should be able to work on **problems 1, 3, and 4 over spring break**. You may not have enough knowledge to do problem 2 until after we come back from break.

Guide to this quiz: For problems that ask you to design an efficient algorithm for a certain problem, your goal is to find the most asymptotically efficient algorithm possible. Generally, the faster your algorithm, the more points you receive. For two asymptotically equal bounds, worst-case bounds are better than expected or amortized bounds. The best solution will receive full points if well written, but ample partial credit will be given for any good solution, especially if it is well written. Bonus points may be awarded for exceptionally efficient or elegant solutions.

Plan your time wisely. Do not overwork, and get enough sleep. Your very first step should be to write up the most obvious algorithm for every problem, even if it is exponential time, and then work on improving your solutions, writing up each improved algorithm as you obtain it. In this way, at all times, you have a complete quiz that you could hand in.

Policy on academic honesty: The rules for this take-home quiz are like those for an in-class quiz, except that you may take the quiz home with you. As during an in-class quiz, you may not communicate with any person except members of the 6.046 staff about any aspect of the quiz, even if you have already handed in your quiz solutions. In addition, you may not discuss any aspect of the quiz with anyone except the course staff until you get them back graded.

This take-home quiz is "limited open book." You may use your course notes, the CLRS textbook, and any of the materials posted on the course web site, but *no other sources whatsoever may be consulted*. For example, you may not use notes or solutions to problem sets, exams, etc. from other times that this course or other related courses have been taught. In particular, you may not use any materials on the World-Wide Web. You probably will not find information in these other sources that will help directly with these problems, but you may not use them regardless.

If at any time you feel that you may have violated this policy, it is imperative that you contact the course staff immediately.

Write-ups: Each problem should be written up separately and submitted to the appropriate turn-in area on Stellar. The Stellar website will have a \LaTeX template for each problem solution for you to use. You need not submit a \LaTeX solution, although we would prefer it, but you must submit your solutions electronically.

Your write-up for 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 asymptotic bounds on the running time your algorithm achieves, including whether they are worst-case, expected, or amortized.

Write your solutions cleanly and concisely to maximize the chance that we understand them. When describing an algorithm, give an English description of the main idea of the algorithm. Adopt suitable notation. Use pseudocode if necessary to clarify your solution. Give examples, draw figures, and state invariants. A long-winded description of an algorithm's execution should not replace a succinct description of the algorithm itself.

Provide short and convincing proofs 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. Do not waste effort proving facts that can simply be cited.

Be explicit about running time and algorithms. For example, don't just say that you sort n numbers, state that you are using merge sort, which sorts the n numbers in $O(n \lg n)$ time in the worst case. If the problem contains multiple variables, analyze your algorithm in terms of all the variables, to the extent possible.

Part of the goal of this quiz is to test your engineering common sense. If you find that a question is unclear or ambiguous, make reasonable assumptions in order to solve the problem. 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.

Good Luck!

Problem 1. For Whom the Road Tolls. [24 points]

Eleanor Sevt is planning to drive from Boston to Los Angeles, but she has only limited funds to make the trip. Fortunately, she is using the MIT solar car, which requires no gas for the journey. Thus, her only costs will be the toll roads she takes on the way. She has acquired a digital model of the North American road network, which consists of a directed graph $G = (V, E)$, where V represents road intersections and E represents the roads themselves. For each edge $e \in E$ there is a length $\ell(e) \in \mathbb{R}$ in miles and a cost $c(e) \in \mathbb{Z}$ in cents for tolls (most of the costs are 0). Give an efficient algorithm to find the shortest path from Boston to L.A. that does not exceed x cents in tolls, where x is given as input.

Problem 2. Rounding a Square Matrix. [24 points]

(NOTE: You may want to wait to work on this problem until after spring break.)

Consider an $n \times n$ matrix $A = (a_{ij})$, each of whose elements a_{ij} is a nonnegative real number, and suppose that each row and column of A sums to an integer value. We wish to replace each element a_{ij} with either $\lfloor a_{ij} \rfloor$ or $\lceil a_{ij} \rceil$ without disturbing the row and column sums. Here is an example:

$$\begin{pmatrix} 10.9 & 2.5 & 1.3 & 9.3 \\ 3.8 & 9.2 & 2.2 & 11.8 \\ 7.9 & 5.2 & 7.3 & 0.6 \\ 3.4 & 13.1 & 1.2 & 6.3 \end{pmatrix} \rightarrow \begin{pmatrix} 11 & 3 & 1 & 9 \\ 4 & 9 & 2 & 12 \\ 7 & 5 & 8 & 1 \\ 4 & 13 & 2 & 6 \end{pmatrix}$$

Give an efficient algorithm to determine whether such a rounding is possible, and if so, to produce the rounded matrix. Be sure to argue that your algorithm is correct.

Problem 3. Hyperjumping to Cloud City. [24 points]

Having successfully evaded Darth Vader's fleet in the asteroid field near the planet Hoth, Han Solo and his crew need to make their way to Cloud City in the Bespin system to meet with Lando Calrissian. After a damage inspection, Chewbacca reports that the hyperdrive in the Millennium Falcon is partially disabled and can only make up to four jumps to hyperspace — a fifth jump would certainly prove disastrous.

Moreover, the control panel for entering jump coordinates is damaged, and the only hyperspace jumps that can be performed are from the database of "Recent Jumps," which contains a large number n of previous jumps. Each jump in the "Recent Jumps" database is an integer triple $(\Delta x, \Delta y, \Delta z)$, as specified using the Empire Coordinate System (ECS). A jump displaces the ship by that the specified number of light years in each dimension. That is, if the ship's current location is (x, y, z) , its position after the jump is $(x + \Delta x, y + \Delta y, z + \Delta z)$.

Han Solo needs to know as quickly as possible if they can make it to Cloud City, and he needs your help to determine how. The Falcon is currently located at $(-21820, 27348, 36072)$ ECS. Find an efficient algorithm to determine if they can reach Bespin, which is located at $(-23252, 35712, 24387)$ ECS, in at most four of the n jumps from the "Recent Jumps" database, and if so, how.

Problem 4. The Price of a Favor [24 points]

Don Corleone is hearing requests for favors during his daughter Connie's wedding. You are one of the many waiting in line to ask a favor of the Don. It is customary that after requesting a favor, you offer the Don a monetary gift. The Don has a different price X for every favor he grants, which his consigliere (trusted adviser) knows implicitly.

It is highly impolite for you to ask outright, "What will the favor cost me?" to find out the Don's price X . Instead, you must place some cash in an envelope and hand it to the consigliere. The consigliere then inspects the envelope. If your offer meets or exceeds X , the consigliere nods and hands the envelope to the Don, who puts it in his pocket and then grants your favor. If the offer is less than X , however, the consigliere takes your offer and puts it in his own pocket and does not allow the Don to be insulted by your puny gift. In this case, your gift is lost to the consigliere, and you must try again with a bigger gift.

Give and analyze a competitive strategy that minimizes the amount of money you must pay to obtain your favor from the Don.

MIT OpenCourseWare
<http://ocw.mit.edu>

6.046J / 18.410J Design and Analysis of Algorithms
Spring 2012

For information about citing these materials or our Terms of Use, visit: <http://ocw.mit.edu/terms>.