Quiz 2

INSTRUCTIONS

This take-home quiz consists of four problems worth a total of 100 points. Your quiz solutions are due **at noon on Friday, April 13, 2012**, uploaded to the Stellar website. Late quizzes will not be accepted unless you obtain a Dean's support or make prior arrangements with the course staff. You must submit your solutions electronically. Also note that there will be no class, office hours, or recitation this week.

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 efficient algorithm possible.** Generally, the faster your algorithm, the more points you receive. In particular, a randomized algorithm that is more efficient than its deterministic counterpart is preferable. For two asymptotically equal bounds, worst-case bounds are better than expected or amortized bounds. The best possible solution will receive full points if well written, but ample partial credit will be given for other correct solutions, especially if they are well written. Bonus points may be awarded for exceptionally efficient or elegant solutions. If you can't solve a problem as stated, try to think of a simplification of it that you can solve. Substantial partial credit will be given for a well stated, meaningful simplification.

Plan your time wisely. Do not overwork, and be sure to get enough sleep. Your very first step would 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 page, 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 has been taught. **You may not use any materials on the World-Wide Web, including OCW.** You probably won't find information in these other sources that will help directly with these problems, but you may not use them regardless. Also, you may not post any questions or comments on the Piazza site.

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: Your solutions are due electronically on the Stellar website by 11:59 am on Friday the 13th of April, 2012. There is no hardcopy submission option for this test. We encourage you to submit your problems typed in LaTeX, since this makes it easier to read and understand your solutions. We have created a LaTeX template for the takehome exam, which is available on the Stellar website.

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 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. 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, 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.

Good Luck!

Problem 1. Bracelet Division [25 points]

A band of k thieves has stolen a bracelet adorned with t different types of jewels. They want to divide the bracelet fairly between them by cutting the bracelet in a small number of places and dividing the parts. For each type $i = 1, 2, \ldots t$, if there are c_i jewels of this type, then each thief wants at least $\lfloor c_i/k \rfloor$ of the jewels.

The thieves ask Professor X for help. The professor models the problem as a continuous problem: the open bracelet is the unit interval, and it is divided into sub-intervals that represent consecutive jewels of each type. The professor finds a way to partition the "continuous bracelet" fairly between the thieves using $(k-1)t$ cuts. However, those cuts do not necessarily correspond to legitimate cuts of the real bracelet, as they may cut across jewels, crushing them.

Given the cuts suggested by the professor, show how to efficiently compute $(k-1)t$ legitimate cuts that divide the actual jewel bracelet between the thieves to their satisfaction.

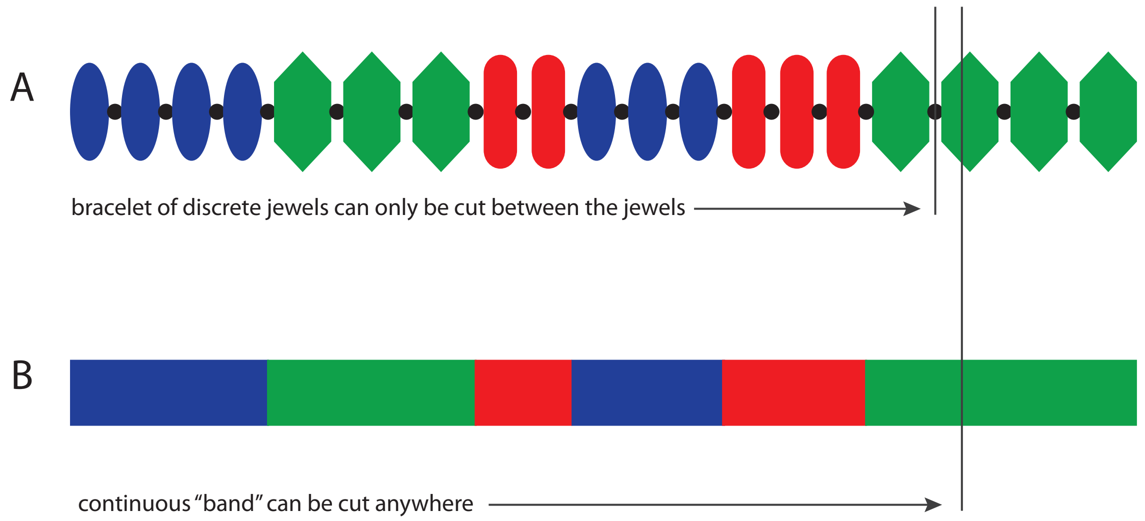
A
bracelet of discrete jewels can only be cut between the jewels
B
continuous "band" can be cut anywhere

Figure 1: Example of (A) a bracelet with jewels and (B) the continuous version of the same bracelet. Notice how a cut across the continuous bracelet may crush a jewel.

Problem 2. Dividing the Delta Quadrant [25 points]

In year 3512, the Star Federation has finally reached a consensus in the century long conflict between Treaps and Cycles for the asteroid field in the Delta Quadrant. To avoid any tension between the two factions, the decision has been made to divide the asteroids in a way that maximizes the minimum distance between any two asteroids assigned to different factions, even if the number of asteroids given to each is uneven (but no faction shall be left without any asteroids).

Design an efficient algorithm which, given a list of n asteroid coordinates (a 3-tuple of real numbers) in the Delta Quadrant, returns the optimal partition maximizing the minimum distance between any two asteroids assigned to different partitions.

Problem 3. A First World Problem [25 points]

Billy Billionaire wants to create an artificial lake in the mountains near his mansion. Fortunately, he lives in 2D land, which makes the problem much easier. He has elevation data, taken at regular intervals across the two-dimensional mountains. We can write this elevation data as $A[1], \ldots, A[n]$. The stretch of land from i to j can be made into a lake if and only if for all $i < k < j$, $A[k] < A[i]$ and $A[k] < A[j]$. We say that the height of the lake $[i, j]$ is equal to $\min\{A[i], A[j]\}$, and the width of the lake $[i, j]$ is equal to $(j - i - 1)$. See Figure 2 for an example.

Billy wants to pick the best stretch of land $[i, j]$ on which to build a lake. He wants the lake to be very wide, so that it looks impressive. However, the lake height should be relatively low, so that his fleet of helicopters have an easier time airlifting the water to the new lake. After some contemplation, Billy decides that he wants to pick a stretch of land $[i, j]$ that can be made into a lake that maximizes $(\text{lakewidth} - \frac{1}{100} \cdot \text{lakeheight})$. Design an efficient algorithm that finds a lake maximizing Billy's chosen objective function.

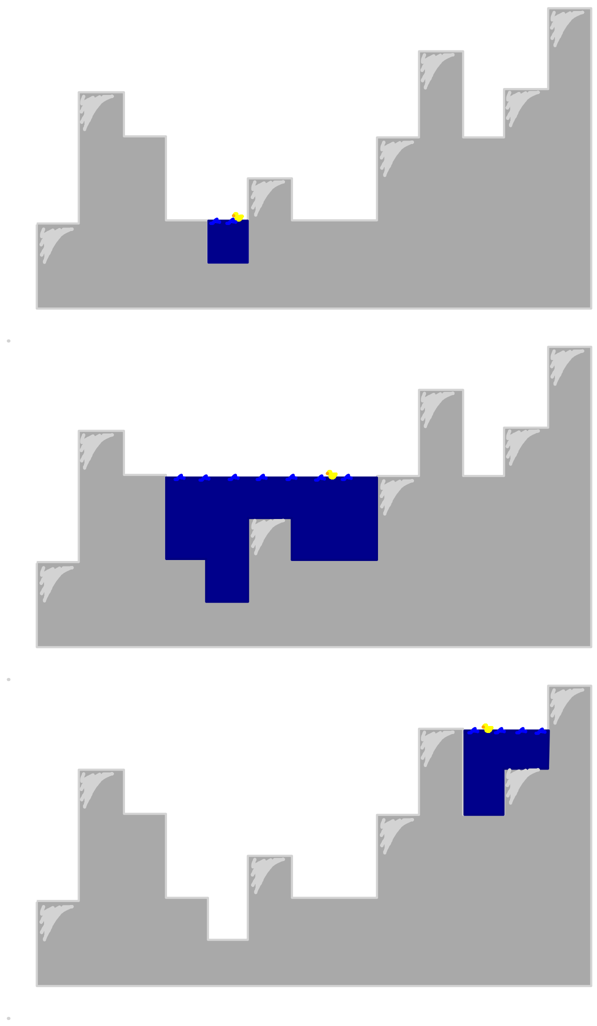

Figure 2: Three different possible lakes on the mountain region described by the array $A = [2, 5, 4, 2, 1, 3, 2, 2, 4, 6, 4, 5, 7]$. The first lake has $\text{lakewidth} = 1$ and $\text{lakeheight} = 2$; the second has $\text{lakewidth} = 5$ and $\text{lakeheight} = 4$; the third $\text{lakewidth} = 2$ and $\text{lakeheight} = 6$.

Problem 4. Social Networking [25 points]

MITbook+ is a new social networking site created by MIT students. Much like other social networking sites, MITbook+ allows you to establish "friendship" with other users. Each user has a unique MITbook+ ID number, and a list of all MITbook+ ID numbers is available to the general public. Furthermore, for any pair of user IDs, there is a publically-accessible webpage that can be used to check whether the pair are friends. However, in the interests of preserving user privacy, all other information associated with MITbook+ IDs is available to members only.

Currently, you are not a user of MITbook+, but you want to learn more about a specific person (known to you by his ID as 60461337). Specifically, you want to determine whether he went to school at MIT. Fortunately, the founders of MITbook+ have posted some statistics on their blog that may help you. Currently, MITbook+ has N users (a very large number). Exactly $2N/3$ users attended MIT; the other $N/3$ did not. Interestingly, because MIT is a very tight-knit community, every current or former MIT student on MITbook+ is friends with every other current or former MIT student on MITbook+.

(a) In the blog post with the statistics, the MITbook+ founders originally claimed that every user on MITbook+ who did not attend MIT was friends with at most $N/4$ users who did attend MIT. Create an efficient algorithm that uses this fact to determine whether user 60461337 attended MIT.

(b) Not long after the original blog post, the founders updated their numbers. Apparently, the original claim spurred a large number of non-MIT users to friend more MIT students. Now, after this friending spree, every user on MITbook+ who did not attend MIT is friends with at most $2N/5$ users who did attend MIT. This also had the effect of raising the average number of friends per user to exactly $2N/3$. Given this updated information, devise an efficient algorithm that checks whether user 60461337 went to school at MIT.