Quiz 2

This take-home quiz contains 6 problems worth 25 points each, for a total of 150 points. **Your quiz solutions are due between 9:30 and 11:00 A.M. on Friday, December 2, 2005, in the Stata lobby (the homework lab).** Late quizzes 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 quiz solutions in person.

The quiz should take you about 12 hours to do, but you have four 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 bounds, the higher your score. If the bounds are the same, a worst-case bound will generally receive a few more points than an expected or amortized bound. Bonus points will be awarded for exceptionally efficient or elegant solutions.

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 during the exam period, even if you have already handed in your quiz solutions. The official end of the exam period is 12:00 noon on Friday, December 2, even though you must hand in your quiz earlier.

This take-home quiz is "limited open book." You may use your course notes, the CLRS textbook, basic reference materials such as dictionaries, 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 have been taught. You may not use materials on the World-Wide Web. These materials will not help you, but you may not use them anyhow.

If at any time you feel that you may have violated this policy, it is imperative that you contact the course staff immediately. If you have any questions about what resources may or may not be used during the quiz, please send email to `6.046 staff`.

Write-ups: Answer each problem on a separate sheet (or sheets) of 3-hole punched paper. Mark the top of each problem with

1. your name,
2. 6.046J/18.410J,
3. the problem number,
4. your recitation time,
5. your TA's name.

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

1. the problem you are solving,
2. the techniques you use to solve it,
3. any important assumptions you make,
4. 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 heapsort, 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.

Bugs, etc.: If you think that you've found a bug, send email to `6.046 staff.` Correct-ions and clarifications will be sent to the class via email and posted on the class website. Check your email and the class website daily to avoid missing potentially important announcements. If you did not receive an email last week reminding you about Quiz 2, then you are not on the class email list and you should let your recitation instructor know immediately.

Survey: Attached to this quiz is a survey on your experiences with the quiz, especially as they relate to academic honesty. Please detach the survey, fill it out, and hand it in when you hand in your quiz solutions. The information you provide will be anonymous. No attempt will be made to identify individuals from their comments. This information will be used to gauge the usefulness of the quiz. Summary statistics and quoted responses will be provided to this and future classes.

PLEASE REREAD THESE INSTRUCTIONS ONCE A DAY DURING THE EXAM.

GOOD LUCK, AND HAVE FUN!

Problem 1. Ups and downs

Moonlighting from his normal job at the National University of Technology, Professor Silvermeadow performs magic in nightclubs. The professor is developing the following card trick. A deck of n cards, labeled $1, 2, \ldots, n$, is arranged face up on a table. An audience member calls out a range $[i, j]$, and the professor flips over every card k such that $i \leq k \leq j$. This action is repeated many times, and during the sequence of actions, audience members also query the professor about whether particular cards are face up or face down. The trick is that there are no actual cards: the professor performs these manipulations in his head, and n is huge.

Unbeknownst to the audience, the professor uses a computational device to perform the manipulations, but the current implementation is too slow to work in real time. Help the professor by designing an efficient data structure that supports the following operations on n cards:

- FLIP(i, j): Flip over every card in the interval $[i, j]$.
- IS-FACE-UP(i): Return TRUE if card i is face up and FALSE if card i is face down.

Problem 2. The Data Center

The world-famous architect Gary O'Frank has been commissioned to design a new building, called the Data Center. Gary wants his top architectural protégé to design a scale model of the Data Center using precision-cut sticks, but he wants to preclude the model from inadvertently containing any right angles. Gary fabricates a set of n sticks, labeled $1, 2, \ldots, n$, where stick i has length x_i. Before giving the sticks to the protégé, he shows them to you and asks you whether it is possible to create a right triangle using any three of the sticks. Give an efficient algorithm for determining whether there exist three sticks a, b, and c such that the triangle formed from them — having sides of lengths x_a, x_b, and x_c — is a right triangle (that is, $x_a^2 + x_b^2 = x_c^2$).

Problem 3. Nonnegativizing a matrix by pivoting

A matrix $M[1 \,..\, n, 1 \,..\, n]$ contains entries drawn from $\mathbb{R} \cup \{\infty\}$. Each row contains at most 10 finite values, some of which may be negative. The goal of the problem is to transform M so that every entry is nonnegative by using only ***pivot*** operations:

```
PIVOT(M, i, x)
for j ← 1 to n
    do M[i, j] ← M[i, j] + x
       M[j, i] ← M[j, i] − x
```

Give an efficient algorithm to determine whether there exists a sequence of pivot operations with various values for i and x such that, at the end of the sequence, $M[i, j] \geq 0$ for all $i, j = 1, 2, \ldots, n$.

Problem 4. Augmenting the Queueinator™

By applying his research in warm fission, Professor Uriah's company is now manufacturing and selling the Queueinator™, a priority-queue hardware device which can be connected to an ordinary computer and which effectively supports the priority-queue operations INSERT and EXTRACT-MIN in $O(1)$ time per operation. The professor's company has a customer, however, who actually needs a "double-ended" priority queue that supports not only the operations INSERT and EXTRACT-MIN, but also EXTRACT-MAX. Redesigning the Queueinator™ hardware to support the extra operation will take the professor's company a year of development. Help the professor by designing an efficient double-ended priority queue using software and one or more Queueinator™ devices.

Problem 5. Spam distribution

Professor Hormel is designing a spam distribution network. The network is represented by a rooted tree $T = (V, E)$ with root $r \in V$ and nonnegative edge-weight function $w : E \to \mathbb{R}$. Each vertex $v \in V$ represents a server with one million email addresses, and each edge $e \in E$ represents a communication channel that costs $w(e)$ dollars to purchase. A server $v \in V$ receives spam precisely if the entire path from the root r to v is purchased. The professor wants to send spam from the root r to $k \leq |V|$ servers (including the root) by spending as little money as possible. Help the professor by designing an algorithm that finds a minimum-weight connected subtree of T with k vertices including the root. (For partial credit, solve the problem when each vertex $v \in V$ has at most 2 children in T.)

Problem 6. A tomato a day

Professor Kerry loves tomatoes! The professor eats one tomato every day, because she is obsessed with the health benefits of the potent antioxidant lycopene and because she just happens to like them very much, thank you. The price of tomatoes rises and falls during the year, and when the price of tomatoes is low, the professor would naturally like to buy as many tomatoes as she can. Because tomatoes have a shelf-life of only d days, however, she must eat a tomato bought on day i on some day j in the range $i \leq j < i + d$, or else the tomato will spoil and be wasted. Thus, although the professor can buy as many tomatoes as she wants on any given day, because she consumes only one tomato per day, she must be circumspect about purchasing too many, even if the price is low.

The professor's obsession has led her to worry about whether she is spending too much money on tomatoes. She has obtained historical pricing data for n days, and she knows how much she actually spent on those days. The historical data consists of an array $C[1 \mathinner{.\,.} n]$, where $C[i]$ is the price of a tomato on day i. She would like to analyze the historical data to determine what is the minimum amount she could possibly have spent in order to satisfy her tomato-a-day habit, and then she will compare that value to what she actually spent.

Give an efficient algorithm to determine the optimal offline (20/20 hindsight) purchasing strategy on the historical data. Given d, n, and $C[1 \mathinner{.\,.} n]$, your algorithm should output $B[1 \mathinner{.\,.} n]$, where $B[i]$ is the number of tomatoes to buy on day i.