SI 335 Spring 2012: Project 3

This is a written project. Solutions to all parts must be typed or very neatly written, and handed in before the beginning of class on **Friday, February 24**. Submissions received by Friday, February 17 will have their grades counted towards the 6-week grades. Late submissions will not be accepted, in accordance with the class policy. Students with excused absences on that date may email a copy to the instructor by the deadline *in addition to handing in a paper copy at the earliest opportunity.*

Guidelines:

- All work must be typed or neatly written
- Multiple-page submissions must be stapled or otherwise bound securely into a single packet.
- **Be concise**. Most problems have at least one simple and relatively short correct answer. Solutions will be graded not only on their correctness but also on how clearly that solution is presented.
- In general, there is a preference for **correct** algorithms over efficient ones. If the problem asks for a $O(n)$ algorithm, a *correct* $O(n \log n)$ algorithm is usually preferable over an incorrect one that claims $O(n)$ running time.
- **Do not try to pass off a solution you know is incorrect**. A statement to the effect of "I know this doesn't work because of blah" will be effective in garnering partial credit points.
- Follow the course honor policy posted here. Remember that **no written collaboration with other humans is permitted**, and that you must document all your sources.

1 Tug of War

There are n Mids trying out for the varsity tug-of-war team. Some of them are strong and some of them are weak. For simplicity, assume that all the strong Mids have *exactly* the same strength, as do all the weak Mids. You can also assume that there is at least one strong Mid and at least one weak Mid. (In particular, n must be at least two.)

Your task is to determine who the weak Mids are, so that they can be kicked off the team. And the only tool you have to determine who is strong and weak is running *contests*. A contest involves pitting some Mids against some others in a tug-of-war, and the outcome can be either that one side wins, or the other side wins, or they tie. There can be any number of Mids in any contest, but it should always be the same number on each side of the contest. The side with more strong Mids (or, equivalently, with fewer weak Mids) wins.

For example, here is an algorithm for $n = 3$ that uses two contests to determine who the weak Mids are:

Input: 3 Mids called A, B, C
Output: A list of the weak Mids

```
1  contest1 := {A} vs {B}
2  contest2 := {B} vs {C}
3  if {A} wins contest1 then
4    -- we know A is strong and B is weak --
5    if {C} wins contest2 then return {B}
6    else return {B,C}
7  else if {B} wins contest1 then
8    -- now we know B is strong and A is weak --
9    if {B} wins contest2 then return {A,C}
10   else return {A}
11 else
12   -- contest1 was a tie --
13   if {B} wins contest2 then return {C}
14   else return {A,B}
```

a) Give a **lower bound** on the number of contests required to determine who the weak Mids are for an input of size n, in the worst case. (Hint: follow the example of the lower bound for sorting from class, and for search in the DQ that followed!)

State your exact lower bound as a function of n, showing all your work. Then state what the asymptotic big-Ω bound is that results, simplified as much as possible.

For example of what I'm asking for, in class we showed that sorting requires at least $\lg(n!)$ comparisons (the exact bound), which is $\Omega(n \log n)$.

b) Give an algorithm for any n that determines who the weak Mids are. In describing your algorithms, you can call the Mids by their number like an array: `M[0]`, `M[1]`, ..., `M[n-1]`.

Analyze the number of contests that your algorithm performs in the worst case (NOT the number of primitive operations, just the number of contests). Then show that your algorithm is asymptotically optimal (you should know what that means).

c) Your algorithm should be asymptotically optimal in the number of contests performed. If the number of contests always EXACTLY matches the lower bound, then fantastic! See part (d).

Otherwise, for this part, state the smallest value of n where your algorithm does more contests in the worst-case than the lower bound says are necessary. For example, the MergeSort algorithm performs 8 comparisons total to sort a size–5 array, but the lower bound only says $\lceil \lg 5! \rceil = 7$ comparisons are necessary.

d) **ULTRA BONUS** Either (1) develop an algorithm whose worst-case number of contests always exactly matches the lower bound from part (a), or

(2) prove that no such algorithm exists for arbitrarily large values of n.

REWARD: For the first person that submits a correct solution to this problem, in your section, I will buy coffee and donuts for everyone and bring them in fresh on the morning of your choice (any day except the midterm). We will not accomplish quite much in class that day. The other section will still have to learn lots of things and listen to me talk the whole time.

2 Ethical Dilemma

This problem will count two times as much as the other problem on this project.

There is a website called `secretbook.com` that offers free accounts to anyone in the world, and allows its members to send secret (encrypted) messages to each other. The way it works is, every user gets assigned a public/private key pair when they sign up for their free account, and the website publishes everybody's public key along with their username and public profile. Then any member can post encrypted messages and files with anyone else's public key, so only that person can read the message.

This website is used by all sorts of people, including political dissidents, underage kids, hackers, professors, and illegal file-sharers. The U.S. government has requested the founders of `secretbook.com` to let them read all the messages, presumably so they can prosecute people sharing illegal things, but the website owners have refused the request.

Here's the problem: you notice that every public key issued by the website has *exactly the same* modulus n. The website owners thought this was OK, since they only issue the public key (e, n) and private key (d, n) to its users, never the prime factors p and q of n or any other information. But you notice there is a vulnerability there...

a) What is the vulnerability? Describe how you would use the setup of the system to gain access to other people's encrypted messages, without being given their private keys. Assume that n has 8192 bits so factoring it would take you an astronomically long time, even with all the computing power in the world. You have to be more clever than that...

b) Regardless of whether you figure out the vulnerability for part (a), take my word that there is one that would allow you to read anyone's encrypted messages on the site. And you seem to have been the first person to notice this!

Now the question is what to do with this information. You have a few options:

- **Tell everybody**. Publish the vulnerability on the web so that anybody can exploit it. Hopefully the website owners are quick!
- **Tell the website owners**. Send a private message to the owners of the website that you have noticed this vulnerability. Maybe you give them an ultimatum (like "fix this or I'll tell more people"), or maybe you kindly request they pay you, or maybe you let them decide.

- **Don't tell anybody**. Keep the vulnerability to yourself, hoping no one else notices. Maybe you read some messages yourself, or maybe you don't. But no one else will know unless they figure it out too.
- **Tell the government**. This leaves the U.S. Government to decide which of the above options to take.

Keep in mind that if you tell anyone, it leaves the possibility that they will use this information in a way that you find unethical. For example, some of the scientists that worked on the Manhattan project were not happy that their work was used to develop a weapon of mass destruction.

Assignment: Write an essay stating what you would do, and why. You should state and develop at least three good supporting arguments for your position. You can draw on historical background, your own ethical or moral reasoning, technical considerations, or (even better!) all of these. You will be judged mostly on the quality of your arguments, but also on standard things like writing style, spelling, and grammar. If you must have a guideline on length, go for 600–800 words, but this is NOT a hard requirement.

Keep in mind that there is not a single right answer — that's why it's called a dilemma! And no weaseling out either, like saying "if such-and-such then I would do X, but otherwise Y, or maybe Z". Based on the information provided in this made-up scenario, make a decision and justify it.