Problem Set 1

Problem 1-1. [0 points] **Instructions**

All parts are due on February 25, 2020 at 10PM. Aim for concise solutions; convoluted and obtuse descriptions might receive low marks, even when they are correct. Solutions should be submitted on Gradescope, and any code should be submitted for automated checking on `https://alg.mit.edu`.

Please abide by the collaboration policy as stated in the course information handout. Note that the first problem is an *Individual* problem and collaboration on this problem is not allowed.

Problem 1-2. [40 points] **Individual problem**

(1) Asymptotic behavior

For each of the following sets of five or six functions, order them so that if f_a appears before f_b in your sequence, then $f_a = O(f_b)$. If $f_a = O(f_b)$ and $f_b = O(f_a)$ (meaning f_a and f_b could appear in either order), indicate this by enclosing f_a and f_b in a set with curly braces. For example, if the functions are:

$$f_1 = n, \qquad f_2 = \sqrt{n}, \qquad f_3 = n + \sqrt{n},$$

then the correct answers are $(f_2, \{f_1, f_3\})$ and $(f_2, \{f_3, f_1\})$.

Note: Recall that a^{b^c} means $a^{(b^c)}$, not $(a^b)^c$, and that $\log$ means $\log_2$ unless a different base is specified explicitly. Stirling's approximation may help for part (d).

(a) [5 points]

$f_1 = (\log n)^{6006}$
$f_2 = 6006^n$
$f_3 = \log(6006n)$
$f_4 = 6006n^{1.001}$
$f_5 = n \log(n^{6006})$

(b) [5 points]

$f_1 = 2^{n^3}$
$f_2 = 3^{2^n}$
$f_3 = 2^{2^{n+1}}$
$f_4 = 2^{2n}$
$f_5 = 4^n$

(c) [5 points]

$f_1 = \log\left((\log n)^3\right)$
$f_2 = n^{3\log n}$
$f_3 = (\log\log n)^3$
$f_4 = (\log n)^{\log(n^3)}$
$f_5 = \log\left(3^{n^3}\right)$
$f_6 = \log\left(\log(n^3)\right)$

(d) [5 points]
Hint: Think about Stirling's approximation for f_5.

$f_1 = 2^n$
$f_2 = 2(n!)$
$f_3 = n^2$
$f_4 = \binom{n}{2}$
$f_5 = \binom{n}{n/2}$
$f_6 = \binom{n}{3}$

(2) Recurrences

Solve each of the following recurrences for $T(n)$. (Use Θ notation to give the asymptotic runtime.) Please justify your answers.

(a) [5 points] $T(n) = 9T(n/3) + n^2$

(b) [5 points] $T(n) = 6006T(n/1.001) + 2^n$

(c) [5 points] $T(n) = 2T(3n/16) + n$

(d) [5 points] $T(n) = T(n/7) + T(11n/14) + \Theta(n)$
Hint: One way to think about this problem is by considering a work tree. First, show that it takes $T(n) = O(n)$ and then that $T(n) = \Omega(n)$.

Problem 1-3. [30 points] **Agent B travels through time.**

Stepbrothers Fineas and Pherb recently fixed up an old time machine they found in the Vandille museum, but before they get a chance to use it, it's stolen by their pet beaver Berry (aka Agent B). Fortunately, he takes it for a good purpose though - stopping the evil Dr. Deinz Hoofenshmirtz.

Agent B knows that, at some point during his life, Dr. Hoofenshmirtz will create a fancy new machine, the Rule-inator, that will allow him to rule the entire tri-state area! He wants to stop Dr. Hoofenshmirtz from using the machine, so he plans to travel through time to destroy it once it is built but before he uses it. Unfortunately, he doesn't know when Dr. Hoofenshmirtz will invent the machine so he's not sure what day he should travel to. However, he does know that the machine takes at least one day to charge, so his goal is to find the machine the day *before* it is used. This is his "target date."

You can assume that whenever Agent B arrives at a new day, he can tell if the machine has been used yet or not. Also assume that the machine will not be used today and that Agent B only needs to destroy the machine once.

(1) [10 points] Agent B assumes that Dr. Hoofenshmirtz has a finite lifespan of at most n more days. Describe and analyze an algorithm that allows Agent B to save the world by using the time machine at most $O(\log n)$ times. Be sure to analyze the correctness and runtime of your solution.

(2) [20 points] It turns out that Dr. Hoofenshmirtz also figured out a way to create a life-extend-inator so now he can live forever! Agent B still needs to find the day he invents the Rule-inator, but he can no longer assume that the Rule-inator will be used within n days from today. Give a way for Agent B to save the world using the time machine at most $O(\log x)$ times, where x is Agent B's target date (the day before the machine is used).

Recall that x is not known to Agent B, so while the runtime of your solution will depend on x, the method Agent B uses cannot depend on knowledge of x. Be sure to analyze the correctness of your solution and justify the number of time-machine uses it involves.

Problem 1-4. [30 points] **Oh, mamma mia!**

Dophie was rummaging around her attic when she found an old diary that her mother Sonna used to keep. She was so excited to read it that she rushed down the stairs and accidentally dropped the diary causing some of the pages to fall out! Dophie knows that the first date in the diary was d, and she wants to figure out the date x of the first missing page. Thankfully, Sonna is known to be organized, so all of her n entries were dated with non-negative integers and sorted by date in ascending order. Also, Sonna wrote exactly one diary entry per day every day, so the dates in her diary are distinct and with no gaps. So, for example, if Sonna's diary entries were originally dated as $[5, 6, 7, 8, 9, 10, 11]$, after Dophie's fall, the diary entries could be $[5, 6, 8, 9, 11]$ where in this case we have $d = 5$ given to us and we see that $x = 7$. Note also that the first entry could be one of the missing ones so, after the fall, the diary entries could be $[6, 8, 10]$ with $d = 5$. In this case, we would have that $x = 5$.

For notation, let D be an array with the current diary entries and d be the date of the first entry to the original diary. As above n is the number of entries in the original diary.

(1) [5 points] Given D, d and j, describe an $O(1)$ time method to determine whether the first date missing is after date j. Be sure to analyze correctness and runtime of your solution.

(2) [12 points] Describe an algorithm to find the date x of the first missing page in $\log(n)$ time.

(3) [3 points] Write and solve the recurrence of the algorithm you proposed.

(4) [10 points] Code and submit.