CSOR S4231 – Summer, 2018

# Homework 1

Out: Saturday, May 26, 2018

Due: 10pm, Tuesday, June 5, 2018

*Please keep your answers clear and concise. For all algorithms* **you** *suggest, you must prove correctness and give the best upper bound that you can for the running time. You should always describe your algorithm clearly in English* **and** *give pseudocode.*

*You should write up the solutions entirely on your own. Collaboration with your classmates is limited to discussion of ideas only. You should list your collaborators on your write-up. If you do not type your solutions, make sure that your hand-writing is legible, your scan is high-quality and your name is clearly written on your homework.*

**You may not use any external resources such as the internet, other textbooks, etc. to solve the homework assignments. Failure to do so will result in receiving a grade of 0 in the assignment and a significantly reduced grade (possibly F) in this class. You should adhere to the department's academic honesty policy (see the course website, too).**

1. (15 points) Show that, if $\lambda$ is a positive real number, then $f(n) = 1 + \lambda + \lambda^2 + \ldots + \lambda^n$ is

   (a) $\Theta(1)$ if $\lambda < 1$.

   (b) $\Theta(n)$ if $\lambda = 1$.

   (c) $\Theta(\lambda^n)$ if $\lambda > 1$.

   Therefore, in big-$\Theta$ notation, the sum of a geometric series is simply the first term if the series is strictly decreasing, the last term if the series is strictly increasing, or the number of terms if the series is unchanging.

2. (20 points) You are given two sorted lists of sizes $m$ and $n$. Give an $O(\log m + \log n)$ algorithm for computing the $k$-th smallest element in the union of the two lists.

3. (20 points) Let $x_1, \ldots, x_n$ be a list of $n$ distinct numbers. We say that $a_i$ and $a_j$ are *inverted* if $i < j$ but $a_i > a_j$. The *Bubblesort* sorting algorithm swaps pairwise adjacent inverted numbers in the list until there are no more inversions (hence the list is sorted).

   Assuming that the input to Bubblesort is a uniformly at random selected permutation of the set $\{x_1, \ldots, x_n\}$, compute the expected number of inversions that need to be corrected by Bubblesort.

4. (25 points) In the table below, indicate the relationship between functions $f$ and $g$ for each pair $(f, g)$ by writing "yes" or "no" in each box. For example, if $f = O(g)$ then write "yes" in the first box. Here $\log^b x = (\log_2 x)^b$.

| $f$ | $g$ | $O$ | $o$ | $\Omega$ | $\omega$ | $\Theta$ |
|---|---|---|---|---|---|---|
| $\log^2 n$ | $25 \log n$ | | | | | |
| $\sqrt{\log n}$ | $(\log \log n)^4$ | | | | | |
| $3n \log n$ | $n \log 3n$ | | | | | |
| $n^{3/5}$ | $\sqrt{n} \log n$ | | | | | |
| $\sqrt{n} + \log n$ | $2\sqrt{n}$ | | | | | |
| $n^2 2^n$ | $3^n$ | | | | | |
| $\sqrt{n} 2^n$ | $2^{n/2+\log n}$ | | | | | |
| $n \log 3n$ | $\frac{n^2}{\log n}$ | | | | | |
| $n!$ | $2^n$ | | | | | |
| $\log n!$ | $\log n^n$ | | | | | |

5. (30 points) In the `MAX SAT` problem, we are given a formula $\phi$ with $m$ clauses over $n$ variables and we want to find a truth assignment that satisfies as many clauses as possible.

Here is a simple randomized algorithm for this problem.

```
for each variable do
    set its value to either 0 or 1 by flipping a coin
end for
```

(a) Suppose that the $j$-th clause has $k_j$ literals. Give the expected number of clauses satisfied by the above algorithm and provide a lower bound for this number in terms of the input parameters.

(b) Next assume that each clause contains exactly $k$ literals. Give the **expected** number of clauses satisfied now and provide a lower bound for this number. How does it compare to the lower bound above?

(c) Now de-randomize the above algorithm as follows: instead of flipping a coin for each variable, select the value that satisfies the most as-yet-unsatisfied clauses. Give a lower bound for the number of clauses satisfied by this deterministic algorithm.

*Hint: let $L_i$ be the number of clauses containing at least one of the variables $\{x_1, \ldots, x_i\}$. Show that after i variables have been assigned, the number of satisfied clauses is greater than $L_i/2$.*

6. (30 points) The Fibonacci numbers are defined by the recurrence

$$F_n = \begin{cases} 0, & \text{if } n = 0 \\ 1, & \text{if } n = 1 \\ F_{n-1} + F_{n-2}, & \text{if } n \geq 2 \end{cases}$$

(a) (4 points) Show that $F_n \geq 2^{n/2}$, $n \geq 6$.

(b) Assume that the cost of adding, subtracting, or multiplying two integers is $O(1)$, independent of the size of the integers.

- (4 points) Write pseudocode for an algorithm that computes $F_n$ based on the recursive definition above. Develop a recurrence for the running time of your algorithm and give an asymptotic lower bound for it.
- (4 points) Write pseudocode for a non-recursive algorithm that asymptotically performs fewer additions than the recursive algorithm. Discuss the running time of the new algorithm.
- (10 points) Show how to compute $F_n$ in $O(\log n)$ time using only integer additions and multiplications.
  (Hint: Express $F_n$ in matrix notation and consider the matrix

  $$\begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}$$

  and its powers.)

(c) (8 points) Now assume that adding two $m$-bit integers requires $\Theta(m)$ time and that multiplying two $m$-bit integers requires $\Theta(m^2)$ time. What is the running time of the three algorithms under this more reasonable cost measure for the elementary arithmetic operations?

**RECOMMENDED EXERCISES (do NOT return, they will not be graded)**

1. Give tight asymptotic bounds for the following recurrences.

   - $T(n) = 4T(n/2) + n^3 - 1.$
   - $T(n) = 8T(n/2) + n^2.$
   - $T(n) = 6T(n/3) + n.$
   - $T(n) = T(\sqrt{n}) + 1.$

2. Problem 7.2 in your textbook, pp. 186-187.

3. Problem 7-4 in the textbook.
   (If necessary, read pages 232-233 to refresh your memory on the definition of a stack.)