

SPRING 2019: COT 5407 INTRO. TO ALGORITHMS

[HOMEWORK 1; DUE JAN 22 AT START OF CLASS]

General submission guidelines and policies: ADD THE FOLLOWING SIGNED STATEMENT. Without this statement, your homework will not be graded.

I HAVE ADHERED TO THE COLLABORATION POLICY FOR THIS CLASS. IN OTHER WORDS, EVERYTHING WRITTEN DOWN IN THIS SUBMISSION IS MY OWN WORK. FOR PROBLEMS WHERE I RECEIVED ANY HELP, I HAVE CITED THE SOURCE, AND/OR NAMED THE COLLABORATOR.

Read the handout on **Homework guidelines and collaboration policy** from your course website before you start on this homework. This is very important.

Problems

1. (Exercise) [Miscellaneous]

- (a) How many bits would I need to write down an integer a in base b ?
- (b) How many bits would I need to write down an integer n^2 in base n ?
- (c) A tree can never be implemented using arrays. True/False? Why or why not?

2. (Exercise) [Big-Oh Notation]

- (a) In the box provided below, write down in big-Oh notation the time complexity of the following code/pseudocode:

Time Complexity of TEST1 is

Algorithm 1 TEST1()

```
1: for  $i = 1$  to  $N$  do
2:   for  $j = i$  down to 1 do
3:     print  $i+j$ 
4:   end for
5: end for
```

- (b) Circle the right choice. To **prove** the claim that $f(n) = O(g(n))$, we need to:
 - i. prove the existence of at least one value of n that satisfies the big-Oh relationship
 - ii. prove the existence of two positive constants that satisfy the definition of the big-Oh relationship.

- iii. prove by induction
 - iv. prove by contradiction
- (c) Circle the right choice. To **disprove** the claim that $f(n) = O(g(n))$, we need to:
- i. prove the existence of at least one value of n that satisfies the big-Oh relationship
 - ii. prove the existence of two positive constants that satisfy the definition of the big-Oh relationship.
 - iii. prove by induction
 - iv. prove by contradiction
- (d) Prove or disprove the following claim:

$$16n = O(n^3)$$

- (e) Prove or disprove the following claim:

$$n^3 = O(16n)$$

Mark the statements that are False for problems (f) through (p). No explanation is required for any of the problems.

- (f) $3n(\log n)^2 + 4n = O(n(\log n)^2)$ for $c = 7$ and $n_0 = 1$.
- (g) $n(\log n)^2 = O(n^2 \log n)$ for $c = 1$ and $n_0 = 1$.
- (h) $n^2 \log n = O(2n^2 \log n + 1)$ for $c = 1$ and $n_0 = 1$.
- (i) If $f(n) = O(g(n))$ and $g(n) = O(h(n))$, then

$$f(n) = O(h(n))$$

- (j) If

$$\lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} = 0,$$

then $f(n) = \Omega(g(n))$ cannot be true because $g(n)$ grows asymptotically faster than $f(n)$.

- (k) $\frac{10}{9} > \frac{5}{4}$.

- (l) For any $x > 0$,

$$\log_x 1 = 0$$

- (m) $k = O(1)$ does not mean that $k = 1$.
- (n) $k = O(n)$ is the same as saying $k = n$.
- (o) For this class, by default (i.e., unless specified otherwise), our main goal is to design algorithms that are efficient for the worst case, not for the average case.

- (p) Name two operations that are efficiently handled by storing items in a RB trees instead of storing it in a sorted array.

3. **(Exercise)** [**Binary Search Trees**]

- (a) Rotations at any node in a Binary Search Tree (BST) will result in another BST, i.e., the BST property is not destroyed by rotations. True or False?
- (b) Explain LEFTROTATE in a BST with the help of a figure. Make sure you label the nodes and subtrees.

4. **(Exercise)** [**Kattis Submission**] The goal here is to learn how to make kattis submissions. Register with kattis using a login name that I can recognize (or let me know what your account ID is). Read the problem called `twostones`, which can be found at <https://open.kattis.com/problems/twostones>

Prepare a programming solution in a language of your choice and learn how to submit it on kattis. The system will not accept your solution if your code has syntax errors or does not solve the problem correctly on its test data. Figure out how you can convince someone else that you have solved the problem (i.e., a screenshot of your profile, or a confirmation of acceptance of your submission).

5. **(Regular)** [**Algorithm Correctness**] Write down pseudocode for the Kattis problem `twostones` mentioned above and argue that your algorithm is correct.