

Analysis of Algorithms
CS 375, Fall 2019

Homework 7

Due **AT THE BEGINNING OF CLASS** Monday, October 7

- From your textbook (CLRS), please read all of Chapters 2 and 3. Although not all of it will show up on CS375 exercises, it is all good material to know.
- *A general note:* When writing up your homework, please write neatly and **explain your answers clearly**, giving all details needed to make your answers easy to understand. Graders may not award credit to incomplete or illegible solutions. Clear communication *is* the point, on every assignment.

Exercises

1. List the following functions according to their order of growth from lowest to highest:

$$(n-2)!, \quad 5\lg(n+100)^{10}, \quad 2^{2n}, \quad 0.001n^4 + 3n^3 + 1, \quad \ln^2 n, \quad \sqrt[3]{n}, \quad 3^n.$$

Although you don't need to explain every part of the ordering for this exercise, please give short explanations (1–2 sentences) for the following:

- (a) how you know the second-smallest comes before the third-smallest; and
 - (b) how you know the second-largest comes after the third-largest.
2. CLRS Exercise 3-4, parts b and f (page 62).
For each part, if the assertion is true, say so, and if not, provide a *counterexample* (i.e., a specific, concrete example that demonstrates the assertion is not true). There is no need to give full proofs of your answers, but please give short explanations (2–3 sentences or so) for each answer, explaining your reasoning (e.g., how you know the assertion is true, or what makes your counterexample a counterexample).
 3. Prof. E. Nigma of the Portland Institute of Technology hired you to analyze the algorithm given here in pseudocode, but as usual, Prof. Nigma neglected to explain what the algorithm does.

```
// Input: A matrix A[0..n-1, 0..n-1] of integers
for i = 0 to n-2 do
    for j = i+1 to n-1 do
        if A[i,j] != A[j,i]
            return False
return True
```

In the above, recall that a matrix is essentially just a two-dimensional array, so $A[i, j]$ might in some languages be written as $A[i][j]$.

- (a) What does this algorithm do? Give an English description of what inputs lead to it returning True and what inputs lead to it returning False. (You do not need to give examples as part of your answer, but you are welcome to include example 2D arrays along with the English description, if it would make your answer clearer.)
- (b) Give an *exact* count of the number of array accesses (count $A[x, y]$ as a single array access) done by this algorithm in the worst case on input of size n , and based on that, give the Θ complexity class for this algorithm. Be sure to explain the details behind your answer: show all work that you did to count those operations and arrive at a summation that expresses the running time; and explain how you got from that summation to the Θ complexity bound, giving the relevant threshold n_0 and leading constants c_1, c_2 used to establish that Θ bound.