

EC330 Applied Algorithms and Data Structures for Engineers Spring 2023

Homework 1

Out: January 25, 2022

Due: February 7, 2023

This homework has a written part and a programming part. Both are due at 11:59 am (by noon) on February 7. You should submit both parts on Gradescope.

*This is an **individual** assignment. See course syllabus for policy on collaboration.*

1. Sums [10 pt]

Provide a closed-form solution to the following problems. Make sure you show the steps.

a) $\sum_{i=1}^{330} \left(\frac{1}{3}\right)^i$

b) $\sum_{i=1}^N (2i^3 + 5i^2 - 7i + 9)$

2. Exponents and Logs [10 pt]

Simplify the following expressions. Make sure you show the steps.

a) $\log_x x^{330x}$

b) $\log_{330}(330^{330} \cdot 330)$

3. Combinatorics [10 pt]

How many integer solutions of $x_1 + x_2 + x_3 = 15$ satisfy $x_1 \geq 4$, $x_2 \geq 2$ and $x_3 \geq -3$?
Make sure you show the steps.

4. Proof [10 pt]

The function $T(n)$ is defined recursively as $T(n) = 2T(n-1) + 1$ where n is a non-negative integer and $T(0) = 1$.

Determine if the following proposition is true.

$$2^n \leq T(n) \leq 2 * 2^n \text{ for all } n \geq 0$$

If you think it is false, provide a counterexample showing that the proposition is false.

If you think it is true, prove that it is true.

5. Program Understanding [10 pt]

a) Describe the behavior of the following function `foo` (e.g. what it returns) for *all possible values* of `x`. Note that its behavior can be different depending on what `x` is.

```
int foo(int x) {  
    if (x==1) return 1;  
    else return 2*foo(int(x/2));  
}
```

- b) What is the value of `sum` after the double-loop exits in the following program? Express your answer as a function of `n`. You can consider `n` as an input to this program. Show your steps.

```
int sum = 1;
for (int i = 2; i < n; i++) {
    sum = sum + (1 << i);
}
```

6. Programming [50 pt]

Make sure to acknowledge any source you consult at the top of your program. Do not include a `main` in your submitted files.

- a) Write a program to generate the result of $fun(x, y)$ for non-negative integer inputs x and y . The function $fun(x, y)$ is defined as follows.

$$fun(x, y) = \begin{cases} 1 & x = 0 \text{ and } y = 0 \\ fun(1, y - 1) & x = 0 \text{ and } y \neq 0 \\ x + 1 & x \neq 0 \text{ and } y = 0 \\ fun(x - 1, y - 1) & \text{otherwise} \end{cases}$$

If your implementation is recursive, it must not make redundant recursive calls, e.g. if the value of $fun(2, 3)$ has been computed before then your program should not compute it again.

Some stub codes have been provided to you in `fun.cpp`. Try to make your algorithm as efficient as you can.

Submit your completed `fun.cpp` file on Gradescope. [20 pt]

- b) Wenchao wants to divide the class into two (non-empty) project groups based on the students' birthdays. In particular, we are going to consider only the day of birth (i.e. an integer between 1 and 31) and not the month or the year. We want the two groups to be "balanced" in such a way that *the difference between the sum of birthdays of one group and the sum of birthdays of the other group is minimized*. For example, say the birthdays for students Alice, Bob, Charlie, Drew and Edward are 3, 27, 4, 5 and 20 respectively. The most balanced group assignment would be {Alice, Bob} and {Charlie, Drew, Edward} since $|(3 + 27) - (4 + 5 + 20)| = 1$ ($||$ indicates taking the absolute value) is the smallest among all possible assignments.

Develop an algorithm to help Wenchao determine the most balanced group assignment for the class. Implement the `balancedGroups` function in `balancedGroups.cpp` and submit this file on Gradescope. [30 pt]

Hint: Think about the different cases for assigning a student to either of the two project groups.