EC330 Applied Algorithms and Data Structures for Engineers
Spring 2021

Homework 1

Out: February 7, 2020
Due: February 15, 2021

This homework has a written part and a programming part. Both are due at 11:59 pm on February 15. You should submit both parts on Gradescope.

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

Quiz 0 counts for 20% of this homework.

1. **Sums [10 pt]**
   Provide a closed-form solution to the following problems. Make sure you show the steps.
   a) $\sum_{i=0}^{\infty}\left(\frac{2}{11}\right)^i$

$$\sum_{i=0}^{\infty}\left(\frac{2}{11}\right)^i = \frac{1}{1-\frac{2}{11}} = \frac{11}{9}$$

   b) $\sum_{i=1}^{N}(i^3 + 2i^2 - 4i + 6)$

$$\sum_{i=1}^{N}(i^3 + 2i^2 - 4i + 6) = \sum_{i=1}^{N} i^3 + 2\sum_{i=1}^{N} i^2 - 4\sum_{i=1}^{N} i + 6\sum_{i=1}^{N} 1$$
$$= \frac{N^2(N+1)^2}{4} + \frac{2N(N+1)(2N+1)}{6} - \frac{4N(N+1)}{2} + 6N$$

Good enough for full credits without further expansion and simplification.

2. **Exponents and Logs [10 pt]**
   Simplify the following expressions. Make sure you show the steps.
   a) $log_x x^{55x}$

$$\log_x x^{55x} = 55x \log_x x = 55x \cdot 1 = 55x$$

   b) $log_{55}(55^{55} \cdot 55)$

$$\log_{55}(55^{55} \cdot 55) = \log_{55} 55^{56} = 56 \log_{55} 55 = 56 \cdot 1 = 56$$

3. Combinatorics [5 pt]

How many integral solutions of $x_1 + x_2 + x_3 = 13$ satisfy $x_1 \geq 5$, $x_2 \geq 0$ and $x_3 \geq -3$?

Let $y_1 = x_1 - 5, y_2 = x_2, y_3 = x_3 + 3 \rightarrow x_1 = y_1 + 5, x_2 = y_2, x_3 = y_3 - 3$

Substitute into the original equation, we get

$y_1 + y_2 + y_3 = 11, y_1 \geq 0, y_2 \geq 0, y_3 \geq 0$

The number of integer solutions can then be found by calculating

$\binom{11+3-1}{3-1} = \binom{13}{2} = 78$

4. Induction [10 pt]

Consider Fibonacci numbers $F_0, F_1, F_2, \ldots$ defined by the following rule

$$F_0 = 1, F_1 = 1, F_n = F_{n-1} + F_{n-2}.$$

In this problem, we will confirm that this sequence grows exponentially fast and obtain bounds on this growth.

a) Use induction to prove that $F_n \geq 2^{0.5n}$ for $n \geq 2$.

b) Find a constant $c < 1$ such that $F_n \leq 2^{cn}$ for all $n \geq 0$.

a) $H(n)$: $F_n \geq 2^{0.5n}$

Base case: $H(2)$: $F_2 = F_1 + F_0 = 2 \geq 2^{0.5 \cdot 2} = 2$

$H(3)$: $F_3 = F_2 + F_1 = 3 \geq 2^{0.5 \cdot 3} \approx 2.828 \ldots$

We want to show that $H(k)$ and $H(k-1)$ true implies $H(k+1)$ is true

$H(k+1)$: $F_{k+1} = F_k + F_{k-1} \geq 2^{0.5k} + 2^{0.5(k-1)} = 2^{0.5k}(1 + 2^{-0.5})$

$\geq 2^{0.5(k+1)}$

Hence, by induction, $F_n \geq 2^{0.5n}$ for $n \geq 2$.

b) One constant that works is c = ¾ (right between 0.5 and 1).

You can actually prove that it works again by induction.

5. Program Understanding [5 pt]

What is the value of `sum` after the double-loop exits in the following program? Express your answer as a function of `n`. Show your steps.

```
int sum = 0;
for (int i = n; i > 0; i--) {
     for (int j = n - i; j < n; j++) {
          sum = sum + 2;
     }
}
```

$$\sum_{i=1}^{n} \sum_{j=n-i}^{n-1} 2 = 2 \sum_{i=1}^{n} (n - 1 - n + i + 1)$$
$$= 2 \sum_{i=1}^{n} i$$
$$= n(n+1)$$

6. Programming [40 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) You are given an array of lower-case letters (e.g., {b, b, x}). Suppose every letter appears even number of times except for one. Write a C++ program that finds this odd-appearing letter. Your program should run in time **O**(n) where n is the size of the input array. **[20 pt]**

   Example #1:
   Input: {b, b, x}
   Output: x
   Example #2:
   Input: {c, b, d, c, c, d, b, b, b}
   Output: c

   Your job is to implement the function *findOdd* in *findOdd.cpp*.
   Submit your solution Gradescope.

b) Write a C++ program that, given an integer, takes the first digit and keeps applying the modulo operation on consecutive digits from left to right such that the result of the module operation will be used on the next digit. If a digit is 0, it should skip that digit (since we cannot divide by 0). **[20 pt]**

   Example:
   Input: 74602
   Output: 1
   Explanation: 7 mod 4 = 3, 3 mod 6 = 3, skip 0, 3 mod 2 = 1.

   Your job is to implement the function *keepMod* in *keepMod.cpp*. Try to optimize your code as much as you can.
   Submit your solution on Gradescope.