

EC330 Applied Algorithms and Data Structures for Engineers Spring 2022

Homework 2

Out: February 8, 2021
Due: February 17, 2021

This homework has a written part and a programming part. Both are due at 11:59 pm on February 17. You should submit both parts on Gradescope.

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

1. Asymptotic Comparison [40 pt]

In each of the following situations, indicate whether $f = O(g)$, or $f = \Omega(g)$, or both (i.e. $f = \Theta(g)$). Justify your choice. [5 pt each]

| | $f(n)$ | $g(n)$ |
|----|---------------------|------------------|
| a) | $n^{1/2}$ | $n^{2/3}$ |
| b) | $330(n + \log n)$ | $n + (\log n)^2$ |
| c) | $330 \log n$ | $\log(n^3)$ |
| d) | $n^{1.01}$ | $n \log^2 n$ |
| e) | $n^2 / \log n$ | $n (\log n)^2$ |
| f) | $(\log n)^{\log n}$ | $n / \log n$ |
| g) | $n 2^n$ | 3^n |
| h) | $\sum_{i=1}^n i^k$ | n^{k+1} |

2. Programming [60 pt]

- a) Write a program that accepts an integer array *nums* and returns the sum *closest* to 330 by adding up *three* integers in this array. For example, if *nums* = [20, 120, 200, 5], then the function should return 325 because $325 = 200 + 120 + 5$ is closer to 330 than $340 = 200 + 120 + 20$. If there is a *tie*, output the *smaller* sum. The function declaration is given below.

`int sumTo330(vector<int> nums);`

Your job is to implement the *sumTo330* function in *sumTo330.cpp*. Submit your solution on Gradescope. You will receive 10 bonus points if your solution runs in $O(n^2)$ time. [30 pt + 10 pt bonus]

- 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 $|(3 + 27) - (4 + 5 + 20)| = 1$ is the smallest among all possible assignments.

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

Hint: Similar to the partition problem that we went over in class, think about the different cases for assigning a student to either of the two project groups.