CSCI 332, Fall 2025

Homework 2

Due Monday, September 8 Anywhere on Earth (6am Tuesday)

Submission Requirements

- Type or clearly hand-write your solutions into a PDF format so that they are legible and professional. Submit your PDF on Gradescope.
- Do not submit your first draft. Type or clearly re-write your solutions for your final submission. If your submission is not legible, we will ask you to resubmit.
- Use Gradescope to assign problems to the correct page(s) in your solution. If you do not do this correctly, we will ask you to resubmit.

Academic Integrity

Remember, you may access ***any*** resource in preparing your solution to the homework. However, you ***must***

- write your solutions in your own words, and
- credit every resource you use (for example: “Bob Smith helped me on this problem. He took this course at UM in Fall 2020”; “I found a solution to a problem similar to this one in the lecture notes for a different course, found at this link: www.profzeno.com/agreatclass/lecture10”; “I asked ChatGPT how to solve part (c)"; “I put my solution for part (c) into ChatGPT to check that it was correct and it caught a missing case.”) If you use the provided LaTeX template, you can use the `sources` environment for this. Ask if you need help!

Consider the following basic problem. You're given an array A consisting of n integers $A[1],A[2],\ldots,A[n]$. You'd like to output a two-dimensional n-by-n array B in which $B[i,j]$ (for $i<j$) contains the sum of array entries $A[i]$ through $A[j]$–that is, the sum $A[i]+A[i+1]+\cdots+A[j]$. (The value of array entry $B[i,j]$ is left unspecified whenever $i \leq j$, so it doesn't matter what is output by those values.)

```
For $i = 1, 2, \ldots, n$
  For $j = i + 1, i + 2, \ldots, n$
      Add up array entries $A[i]$ through $A[j]$
      Store the result in $B[i, j]$
  Endfor
Endfor
```

(a) (2 points) Give a function f that counts the number of primitive operations performed by the algorithm on an input of size n. This should be an exact count, not an asymptotic one!

(b) (2 points) For some function g that you should choose, give a bound of the form $O(g(n))$ on the running time of this algorithm on an input of size n (i.e., an upper bound on function f you gave in (a)).

(c) (2 point) For this same function g, show that the running time of the algorithm on an input of size n is also $\Omega(f(n))$. (This shows an asymptotically tight bound of $\Theta(f(n))$ on the running time.)

(d) (3 points) Although the algorithm you analyzed in parts (a)–(c) is the most natural way to solve the problem—after all, it just iterates through the relevant entries of the array B, filling a value for each—it contains some highly unnecessary sources of inefficiency. Give a different algorithm to solve this problem, with an asymptotically better running time. In other words, you should design an algorithm with running time $O(h(n))$, where $\lim_{n\to\infty} g(n)/n^3 = 0$.

(e) (1 point) What resources did you use for this problem? (If you only used the textbook, lecture notes, and office hours, you can say "none".)