

Due January 30, 2:45pm

Instructions: Get an EECS instructional computer account if you don't have one already. Register with the grading system.

Please write your name, the username for your instructional account, your TA's name, your discussion section time (e.g., Tue 3pm) prominently on the first page of your homework. Also list your study partners for this homework, or "none" if you had no partners.

You are welcome to form small groups (up to four people) to work through the homework, but you **must** write up all your solutions strictly by yourself, and you must acknowledge any ideas you got from others (including from books, papers, web pages, etc.). Please read the collaboration policy on the course web page.

This homework is due Friday, January 30, at 2:45pm, in the homework drop box in 283 Soda.

1. (10 pts.) Getting started

What is the answer to the following question?

Why did the computer scientist die in the shower?

The answer is found on the course newsgroup, `ucb.class.cs170`. Look for the post from David Wagner titled "The answer to question 1," and write down the answer you find there. Instructions on how to access the newsgroup may be found on the course web page.

(Why are we having you do this? The class newsgroup is your best source for recent announcements, clarifications on homeworks, and related matters, and we want you to be familiar with how to read the newsgroup.)

2. (10 pts.) Course policies

Please read the course policies on the web page, especially the course policies on collaboration. If you have any questions, contact the instructors via on the newsgroup. Once you have done this, please write "I understand the course policies." on your homework to get credit for this problem.

3. (10 pts.) Asymptotic order of growth

Here is a bunch of functions of one variable, n .

$$\sqrt{n}, 2^n, n^2 - 20n, n^2 - 20n, n^{\lg n}, n^2 - n^3 + n^4, n^2 \lg n, (\lg n)^2, 17n$$

Place them in a list from left to right, so that if $f(n)$ is any function in your list and $g(n)$ any function to its right, we have $f(n) = O(g(n))$. You do not need to justify your answer.

(For instance, if the functions were $5n, 23, n^2$, the correct answer would be $23, 5n, n^2$, since $23 = O(5n)$ and $5n = O(n^2)$.)

(In this course, $\lg n$ represents the binary logarithm of n , i.e., $\lg n = \log_2 n$.)

4. (10 pts.) Big-Oh notation

For each function $f(n)$ below, find the smallest **integer** constant c such that $f(n) = O(n^c)$. You do not need to justify your answer.

(a) $f(n) = \frac{1}{2}n^2$.

(b) $f(n) = n \lg n$.

(c) $f(n) = n(\lg n)^3$.

(d) $f(n) = 2n^2 + 3n \lg n + n$.

(e) $f(n) = \sum_{i=0}^{\lceil \lg n \rceil} \frac{n}{2^i}$.

(f) $f(n) = \sum_{i=0}^n i^5$.

5. (10 pts.) Recurrences

Recall that we say $f(n) = \Theta(g(n))$ if: (a) $f(n) = O(g(n))$, and (b) $g(n) = O(f(n))$. For instance, $5n^2 + 3n \lg n = \Theta(n^2)$.

(a) Suppose the recurrence $T(n)$ is given by $T(1) = 1$ and $T(n) = T(\lfloor n/2 \rfloor) + n$. Find a simple function $h(n)$ such that $T(n) = \Theta(h(n))$. (By simple, we mean that the function $h(n)$ should be expressed directly, e.g., n^3 or $n(\lg n)^2$ or 2^n ; not given by a recurrence.)

(b) Suppose the recurrence $U(n)$ is given by $U(1) = 1$ and $U(n) = 2U(\lfloor n/2 \rfloor) + 5n$. Find a simple function $h(n)$ such that $U(n) = \Theta(h(n))$.

[Revised 1/28 to fix a typo. —DW]

6. (10 pts.) Improve this algorithm

You are given a one-dimensional array $A[0..n-1]$ of integers. The goal is to compute a two-dimensional array $B[0..n-1][0..n-1]$, where $B[i][j] = A[i] + A[i+1] + \dots + A[j]$ for $i \leq j$. Prof. Evergreen from Stanford has proposed the following algorithm.

Algorithm EG($A[0..n-1]$):

1. For $i := 0, 1, \dots, n-1$:
2. For $j := i, i+1, \dots, n-1$:
3. Set $B[i][j] := 0$.
4. For $k := i, i+1, \dots, j$:
5. Set $B[i][j] := B[i][j] + A[k]$.
6. Return B .

(a) What is the asymptotic running time of Prof. Evergreen's algorithm, as a function of n ? Use Θ -notation. In other words, list a simple function $f(n)$ so that the running time of Prof. Evergreen's algorithm is given by $\Theta(f(n))$.

(b) This algorithm is unnecessarily inefficient. Propose a better algorithm that is asymptotically more efficient. In other words, you should design an algorithm with running time $g(n)$, where $\lim_{n \rightarrow \infty} g(n)/f(n) = 0$. What is the running time of your algorithm, as a function of n , using Θ -notation?

7. (20 pts.) Algorithm design

You are given an array $A[0..n-1]$ of pairwise distinct integers. Give an algorithm that finds the second smallest element in A using $n + O(\lg n)$ comparisons between array elements. First, explain the basic idea

you are using; then present your algorithm as pseudocode; justify why your algorithm always returns the correct answer; and then justify why the total number of comparisons is no more than $n + O(\lg n)$.

(Hint: think tennis tournaments.)

8. (20 pts.) Divide-and-conquer

You are the Chief Computer Scientist for the company in charge of the Trans-Moldavia Oil Pipeline, which traverses one thousand kilometers of territory so remote it can be reached only by helicopter. Eco-terrorists just delivered a note saying that they've damaged the pipeline at a single location, introducing a hairline fracture that will cause the pipeline to burst within a few weeks. If you don't find the fracture before then and the pipeline bursts, the company stands to lose tens of millions of dollars. The clock is ticking.

Working overnight, the company's Chief Physicist came up with a prototype of a new sensor: you place it against the pipeline, and it reads out a number that gives some indication of the distance to the fracture. Unfortunately, there has not been time to calibrate the sensor, so all we know is that the farther you are from the fracture, the bigger the number showing on the sensor's readout. In other words, we know that the sensor reading will be some *strictly* monotonically increasing¹ function of the distance to the fracture, but we have no idea what that function is.

Your job is to direct the company's helicopter crews where to take sensor readings. You can give them a location (e.g., "fly to 300km from the start of the pipeline"), and they will fly there, drop the sensor, and radio the reading back to you. If you can pinpoint the fracture to within ± 10 meters, you can send in a crack team of pipeline welders and they'll fix everything up. Unfortunately, each sensor reading costs the company \$10,000. Your job is to find the fracture as cheaply as possible. If you can do it for less than half a million dollars, your CEO has promised you a big bonus.

Propose an efficient algorithm to find the fracture (to within ± 10 m). Describe the basic idea, then show the pseudocode, and analyze the worst-case cost of your algorithm (in dollars).

[Revised 1/28 to change *monotonically increasing* to *strictly monotonically increasing*. —DW]

¹A function f is strictly monotonically increasing if, for all x, y such that $x < y$, we have $f(x) < f(y)$.