CSCI 332, Fall 2024

Homework 7

Due before class on Tuesday, October 22, 2024–that is, due at 9:30am Mountain Time

---

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.
- 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.
- You may work with a group of up to three students and submit ***one single document*** for the group. Just be sure to list all group members at the top of the document. When submitting a group assignment to Gradescope, only one student needs to upload the document; just be sure to select your groupmates when you do so.

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!

Grading

Remember, submitted homeworks are graded for completeness, not correctness. Correctness is evaluated using homework quizzes.

Each submitted problem will be graded out of six points according to the following rubric:

- Does the solution address the correct problem?
- Does the solution make a reasonable attempt at solving the problem, even if not fully correct?
- Is the presentation neat?
- Is the explanation clear?

- Does the solution list collaborators or sources, or state that the student did not use any collaborators or outside resources?
- Is the solution written in the student's own voice (not copied directly from an outside resource)?

1. You work for the State of Montana's Economic Development Office, and you have been tasked with analyzing income data from Montana's households. However, the data source that you must use is a bit complicated. In order to save money, the state contracted with two different companies to collect income data in Montana's two congressional districts, which we will assume each have exactly n households. We will also assume that no two Montana households have exactly the same income.

   The two companies worked independently and use different databases to store their results. Each database stores income data for their respective n Montana households, so overall, there is data for $2n$ Montana households across these two databases. These databases support one query: given an integer k, they will return the kth smallest income.

   The companies did the data collection for free, but now charge $1 for every query to the database.

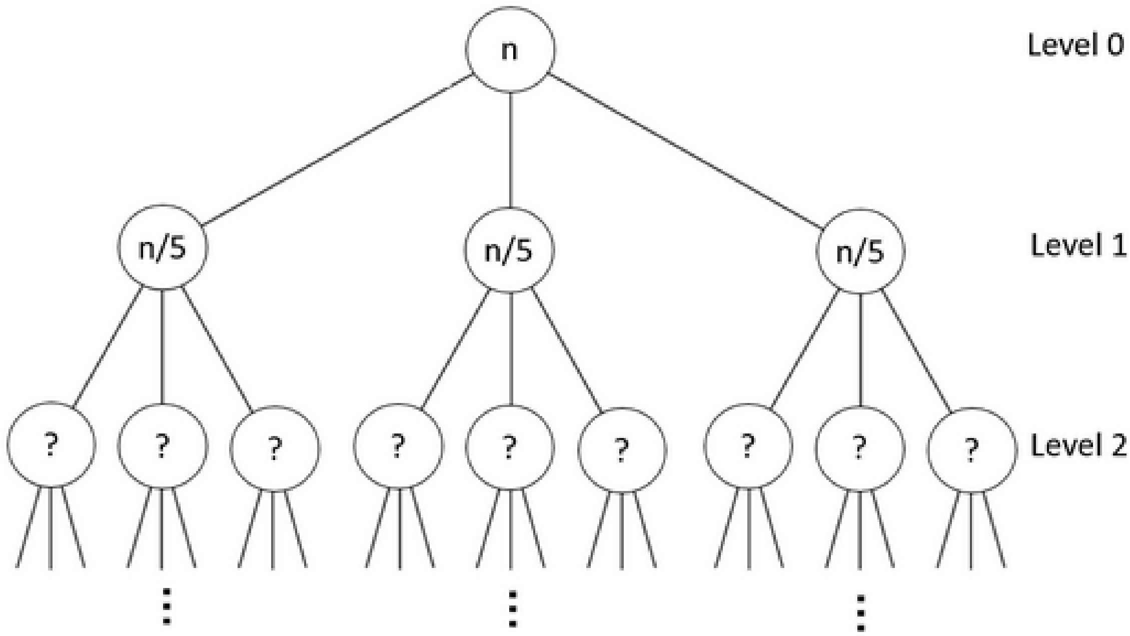
   You want to figure out the *median* household income—that is, the n^{th} household income in the combined, sorted list.

   Your goal is to give algorithm to find the median value using $O(\log n)$ queries, and thus costing $O(\log n)$ dollars.

   Some hints:

   - Remember that this is the divide and conquer chapter! Your goal is to solve this by dividing the input and solving recursively.
   - If you want to write pseudocode, you'll probably want to use a function to query the database. If it helps, you can assume that the function `query(`D_i`, k)` returns you the kth smallest element from database i in constant time. So you will want to make calls like `query(`D_1`, `$n/2$`)` to indicate that you're querying database 1 with the value $n/2$.
   - If you like, you can assume that n is a power of two. (This will mean that if you divide n in half, you'll always get a whole number.)
   - Finally, a tricky thing about this problem is that you can't manipulate the databases, you can only query them. In particular, you can't delete half of the database and then pass the remaining half to your next recursive call. Instead, you'll have to pass around indices into the databases.

2. Consider the recursion tree below, which represents the recursive calls (each node) and size of the input to the recursive calls (text inside each node).



In general, we can write a recurrence relation as

$$T(n) = aT(n/b) + f(n)$$

where $T(n)$ is the runtime of the algorithm on an input of size n and $f(n)$ is the non-recursive part of the algorithm (i.e., any preprocessing before or postprocessing after the recursive call(s)).

(a) What are a and b for this recurrence relation?

(b) What is the depth of the recursion tree, as a function of n?

(c) Assume that $f(n) = n^2$. At the following levels of the recursion tree, how many computations are being done over all calls at that level?

  (i) Level 0

  (ii) Level 1

  (iii) Level 2

  (iv) Level 3

(d) Based on your answer to (c), make a guess about an asymptotic bound on $T(n)$. Either give an $f(n)$ such that $T(n)$ is $O(n)$ and explain why you aren't sure whether there is a smaller such f, or give a $g(n)$ such that $T(n)$ is $\Theta(g(n))$ and explain why you think this is a tight bound.