Jeff Ondich
Fall 2022

Due Dates: One question of your choice (either #1 or #2) is due by **11:59PM Sunday, 18 September** and will be graded on completion only (1 point). Your full answers to all questions are due by **11:59PM Wednesday, 21 September**. Each solution will be graded for correctness and clarity (4 points per question). Read the course information page for further details.

At the top of your write-up for each problem, please estimate the amount of time you spent on the problem, list your collaborators, and briefly describe the nature of your collaboration. This information will help me calibrate the difficulty of future problem sets. Thanks!

1. Guess what? You have been hired at the US Internal Revenue Service (IRS) as a data analyst. You have a dataset consisting of a single column of numbers, unsorted. This column of numbers is the gross adjusted income, rounded to the nearest 1000 dollars, from every US individual tax return filed for the 2021 tax year. So for example, there might be a 42 in the column representing an entry-level marketing associate who made $41,850 last year, a 1100 representing the $1.1 million made by the president of Harvard, and a 4400000 for Jeff Bezos.

   Your bosses want you to do a bunch of modeling experiments to imagine the impact of various proposed tax policies. Based on the many, many feature requests handed to you from up above, you have decided that there's one computational operation you need to make really fast: given a salary range (e.g., $50,000 - $75,000), how many people (or, more accurately, how many tax returns) are there in that range? We're going to call this count the *range-count* for the specified range.

   Of course, the brute force approach to this problem is to scan through the entire column of numbers and count how many of them are in the specified range. The problem is that there are about 260 million tax returns, and you need to perform this operation a *lot*, with a ton of different randomized salary ranges, to complete your project. So an $O(n)$ algorithm just won't cut it.

   More technically, here's the situation. You have a list L containing n integer incomes in the range $[0, m]$. (Here, *income* means *the gross adjusted income from one of the tax returns, expressed as an integer number of thousands of US dollars.*) Your goal is to devise an algorithm to compute the range-count for any income range in $O(1)$ time.

   You may ***preprocess*** L as long as you can do it in $O(n + m)$ time. Preprocessing is a one-time action done before you run your range-count algorithm. You will then run range-count a zillion times (where *zillion* is a technical term whose definition is *I dunno for sure, but a lot*).

   Here's range-count:

   Input: two integers $x \in [0, m]$ and $y \in [0, m]$ where $x \leq y$.
   Output: the number of incomes in the range $[x, y]$.

   For example, suppose $L = \langle 30, 25, 90, 80, 23, 35, 75, 70 \rangle$. Given $x = 25$ and $y = 75$, range-count would be 5.

   In your solution, make sure you clearly describe both your pre-processing step and how you determine the number of incomes in the specified range. You must also provide a proof (i.e., a well-reasoned and clearly articulated argument) that your approach is correct and meets all the specifications.

2. (This problem has been retained verbatim from Layla, in honor of her love for her cat, Yawgoo.) Supposed we live in a world where there are only three kinds of house pets: Cats, Dogs, and Birds.

   You are an in-demand pet training specialist. You have designed three separate training regimens for the three classes of pets.

   To make your preparation easier, you intend to do your training successively to each member of one category of pet, then to the next category, etc., instead of going back and forth among the categories. (But you don't care about the orders within a group. All Cats are the same, just like all Birds, and all Dogs.)

   You are given an array $A[1...n]$ of the n pets on your training list. Each pet $A[i]$ has a type $t(A[i]) \in \{\text{cat, dog, bird}\}$. You are asked to rearrange the array A so that all of the Cats come first, then all the Birds, then all the Dogs. In your solution, you are only allowed two very specific operations on the array:

   (i) for a particular i, query $t(A[i])$; and

   (ii) for two particular indices i and j, swap $A[i]$ and $A[j]$.

   For example, you cannot copy elements from the array to an auxiliary array or even create another array. If you wish, you may have pointers that store indicies into the array. Give an algorithm that runs in time $O(n)$ to perform your assigned task. As always, you must also provide a proof that your approach is correct and obtains the specified runtime.

Follow-up questions

Don't want to stop thinking about this? Here are a couple things to think about if you're interested in thinking further. These aren't worth any points, but if you feel like adding them to the end of your submission, that's cool. (I'm especially interested in your ideas for the second item below.)

- In problem #1, think about putting this into practice. How big might m be? How much memory would be taken up by your preprocessed data structure? (Assuming, say, that each integer in your data structure takes up four bytes.)
- Consider problem #2. This is, essentially, a sorting exercise under restricted circumstances (only three sort keys, and sorting within each sort key is not required). But the birds/cats/dogs context is not realistic, since even if the pet training setup takes a few minutes, the sorting would not–a real pet trainer could easily sort the dozen or so animals per day by hand in a couple seconds. Can you come up with a more realistic context in which this problem might arise?

Jeff Ondich

Submission Logistics

You will write up your solutions to each question in this problem set and typeset it using LaTeX. Convert the LaTeXinto a PDF file and upload it to Moodle.

This will be the default way to submit homework this term. Occasionally, I might ask for a different kind of submission (e.g., a Python program or a link to some Google Slides), but most of the time, you'll do your write-ups in LaTeX.

Use this LaTeXtemplate as the basis for your write-up.

Please see this resources page for more LaTeXhelp.

Need help?

Don't hesitate to use Slack #questions to ask a question. Chances are someone else is having the same problem and will also benefit from your asking.

Adapted from an assignment by Layla Oesper