# CS 170 Homework 2

Due **Monday 9/11/2023, at 10:00 pm (grace period until 11:59pm)**

## 1 Study Group

List the names and SIDs of the members in your study group. If you have no collaborators, you must explicitly write "none".

## 2 Two sorted arrays

You are given two sorted arrays of integers, each of size $k$. Give an efficient (i.e., better than $O(k)$-time) algorithm to find the $k$-th smallest element in the union of the two arrays. You may assume that all the elements are distinct.

Your solution should contain a description of the algorithm, a proof of correctness and runtime analysis. (i.e a 3-part solution). **In addition, please provide corresponding pseudocode.**

## 3 Counting multiples of 3

You are given an array $A$ of $n$ distinct non-negative integers. Give an $O(n(\log n)^2)$ time algorithm to count the number **odd**-sized subsets of $A$ whose elements add up to a multiple of 3.

You can assume that multiplying two $k$ bit integers can be done in $O(k \log k)$ time (see Harvey and van der Hoeven).

*Hint 1: Note that the number of such subsets can scale exponentially with $n$, so we cannot assume that arithmetic operations take $O(1)$ time. To bound the cost of adding and multiplying, show that the number of subsets of a size $n$ array can be stored in an $n$-bit integer.*

*Hint 2: Try keeping track of the number of odd and even sized subsets that sum to 0, 1, or 2 mod 3 using divide and conquer. How do you combine the subproblems of size $k$ in $O(k \log k)$ time?*

## 4 The Resistance

We are playing a variant of The Resistance, a board game where there are $n$ players, $s$ of which are spies. In this variant, in every round, we choose a subset of players to go on a mission. A mission succeeds if the subset of the players does not contain a spy, but fails if at least one spy goes on the mission. After a mission completes, we only know its outcome and not which of the players on the mission were spies.

Come up with a strategy that identifies all the spies in $O(s \log(n/s))$ missions. **Describe your strategy and analyze the number of missions needed**

# 5 Werewolves

You are playing a party game with $n$ other friends, who play either as werewolves or humans. You do not know who is a human and who is a werewolf, but all your friends do. There are always more humans than there are werewolves.

Your goal is to identify one player who is certain to be a human.

Your allowed 'query' operation is as follows: you pick two players as partners. You ask each player if their partner is a human or a werewolf. When you do this, a human must tell the truth about the identity of their partner, but a werewolf doesn't have to (they may lie or tell the truth about their partner).

Your algorithm should work regardless of the behavior of the werewolves.

(a) For a given player $x$, devise an algorithm that returns whether or not $x$ is a human using $O(n)$ queries. Just an informal description of your test and a brief explanation of why it works is needed.

(b) Show how to find a human in $O(n \log n)$ queries (where one query is taking two players $x$ and $y$ and asking $x$ to identify $y$ and $y$ to identify $x$).

*Hint: Split the group into two groups, and use part (a). What invariant must hold for at least one of the two groups?*

**Give a 3-part solution.**

(c) (**Optional, not for credit**) Can you give a $O(n)$ query algorithm?

*Hint: Don't be afraid to sometimes 'throw away' a pair of players once you've asked them to identify their partners.*

**Give a 3-part solution.**

## 6 [Coding] Quickselect

For this week's homework, you'll implement the quickselect algorithm in a python jupyter notebook called `quickselect.ipynb`. There are two ways that you can access the notebook and complete the problems:

1. **On Local Machine**: `git clone` (or if you already have it, `git pull`) from the coding homework repo,

   `https://github.com/Berkeley-CS170/cs170-fa23-coding`

   and navigate to the `hw02` folder. Refer to the `README.md` for local setup instructions.

2. **On Datahub**: Click here and navigate to the `hw02` folder if you prefer to complete this question on Berkeley DataHub.

Notes:

- *Submission Instructions:* Please download your completed `quickselect.ipynb` file and submit it to the gradescope assignment titled "Homework 2 Coding Portion".
- *OH/HWP Instructions:* Designated coding course staff will provide conceptual and debugging help during office hours and homework parties.
- *Academic Honesty Guideline:* We realize that code for some of the algorithms we ask you to implement may be readily available online, but we strongly encourage you to not directly copy code from these sources. Instead, try to refer to the resources mentioned in the notebook and come up with code yourself. That being said, we **do acknowledge** that there may not be many different ways to code up particular algorithms and that your solution may be similar to other solutions available online.