CIS 1210 — Data Structures and Algorithms
Homework Assignment 3

Assigned: September 17, 2024 **Due:** September 23, 2024

Note: The homework is due **electronically on Gradescope** on September 23, 2024 by 11:59 pm ET. For late submissions, please refer to the Late Submission Policy on the course webpage. You may submit this assignment up to 2 days late.

A. **Gradescope**: You must select the appropriate pages on Gradescope. Gradescope makes this easy for you: before you submit, it asks you to associate pages with the homework questions. Forgetting to do so will incur a 5% penalty, which cannot be argued against after the fact.

B. **LaTeX**: You must use the LaTex template provided on the course website, or a 5% penalty will be incurred. Handwritten solutions or solutions not typeset in LaTex will not be accepted.

C. **Solutions**: Please write concise and clear solutions; you will get only a partial credit for correct solutions that are either unnecessarily long or not clear. Please refer to the Written Homework Guidelines for all the requirements.

D. **Algorithms**: Whenever you present an algorithm, your answer must include 3 separate sections.

1. A precise description of your algorithm in English. No pseudocode, no code.
2. Proof of correctness of your algorithm
3. Analysis of the running time complexity of your algorithm

E. **Collaboration**: You are allowed to discuss **ideas** for solving homework problems in groups of up to 3 people but *you must write your solutions independently*. Also, you must write on your homework the names of the people with whom you discussed. For more on the collaboration policy, please see the course webpage.

F. **Outside Resources**: Finally, you are not allowed to use *any* material outside of the class notes and the textbook. Any violation of this policy may seriously affect your grade in the class. If you're unsure if something violates our policy, please ask. If you would like to cite from CLRS in your proofs you must first ask for permission on ED.

1. [10 pts] Divek the Draft King

Divek is constantly bragging about how he won so much in fantasy football last year. The CIS 1210 TAs are dying to know Divek's secret before they form their own teams this season. Divek tells his disciples he will only reveal his strategy if they can guess k, the amount of money he won. The TAs can guess any integer amount of money and Divek will tell them whether their guess is too high, too low, or exactly correct. To make it a bit more entertaining for himself, Divek only allows $O(\lg k)$ guesses.

With the knowledge that k is a positive integer, the TAs must determine how much Divek won before the current season starts! Note that they cannot guess in terms of k. Provide an algorithm that will allow the TAs to figure out Divek's winnings in $O(\lg k)$ guesses. Be sure to justify the correctness and runtime of your algorithm.

2. [15 pts] Connor's Quarterbacks

Connor is competing in the CIS 1210 TA Fantasy Football league and the CIS 1210 students are tasked with predicting what players will be on his team. Connor doesn't care for defense, and instead likes to focus on offense. Disregarding standard fantasy football rules, Connor has a list Q of n quarterbacks, each with a unique QBR (quarterback rating). He has a unique strategy where he drafts x quarterbacks with a QBR closest to the median QBR. You may assume that $x < n$, that n is odd, and that any ties are broken arbitrarily.

Connor is too lazy to sort the list of quarterbacks by their QBRs, so he needs you to give him an $O(n)$ algorithm that he can use to obtain this list of x "median quarterbacks" and analyze its runtime. **No proof of correctness is required.**

As an example, if Q contains five quarterbacks, q_1, q_2, q_3, q_4, q_5, with QBRs given by (42, 14, 2, 90, 34), and $x = 3$, then your algorithm should return q_1, q_2, and q_5.

3. [20 pts] Rohacs the Researcher

Daniel Rohacs has won every CIS 1210 Fantasy Football League since '08, and he intends to keep it that way. He spends hours before each draft on ESPN.com, carefully researching and planning out the best athletes to draft into his franchise.

In his search for a reliable team defense, Daniel has narrowed his analysis down to n teams that he sees potential in. The n teams are arranged in a line on his whiteboard and each has a integer "Yards Allowed per Play" (YAP) value associated with it. YAP represents the average number of yards a defense gives up to the opposing offense on each play. Note that the integers are not necessarily distinct and not necessarily positive. Some of these defenses are so good that they have negative YAP, gaining more yards than they lose across plays.

Daniel wants to select a particular non-empty sequence of consecutive teams such that the product of their YAP is minimized. By sequence of consecutive teams, we refer to the teams being consecutively positioned in the line.

(a) Show that it is possible for Daniel to find the best sequence of consecutive teams that includes the first team in the line in $O(n)$ time by providing an algorithm that does so. The optimal solution is not necessarily unique, but must include the team at the first index. **No proof of correctness is necessary.**

(b) Give an algorithm for finding the overall best sequence of consecutive teams in $\Theta(n^2)$ time. **No proof of correctness is necessary.**

(c) Daniel discovers that Dhruv, who is also really good at fantasy football, has already found a solution to part (b). To help Daniel one up Dhruv, design a $\Theta(n \lg n)$ divide and conquer algorithm that returns the product of the overall best sequence. For this question only, be sure to justify the correctness and runtime of your algorithm. **Solutions that do not use divide and conquer will receive no credit.**

4. [25 pts] Taking Down the Anti-Purdy Propaganda Posse

Leah Ning is a massive fan of Brock Purdy. When she learned that Brock Purdy was chosen as the very last pick in the NFL draft, she was curious to know who the favorite player in Brock Purdy's rookie class was for each of her fellow TAs. Each of the n TAs wrote down the name of their favorite draft pick from Brock Purdy's rookie year of 2022. All TAs have a unique favorite pick.

Upon reading through the TAs picks, Leah realized that each of the TAs had picked a different player and shockingly none of them picked Brock Purdy! In order to investigate how the TAs could dismiss such a promising talent, Leah must devise a strategy to discover the correspondence between TAs and their favorite draft picks.

Reluctant to admit to their foolish defamation of Brock Purdy's name, the rest of the TAs are unwilling to help Leah with her search. **The only way Leah will get information out of any TA is by presenting them with a certain player, to which the TA will tell her whether that player is indeed their favorite pick, or whether their favorite player was drafted before or after, in the pick order**. Additionally, Leah does not have any internet access and cannot directly compare two players to find out who was drafted earlier.

Given all this information, help Leah design an efficient algorithm to discover who is out to tarnish Brock Purdy's legacy!

(a) Leah first proposes the following algorithm to associate TAs with their favorite pick. Let T denote the group of TAs and S be the set of players. What is the *expected* running time of this algorithm? What is its *worst case* running time? Justify your answer.

```
PlayerSorter(T, S):
if |T| = |S| = 1 then:
    Match the player to the TA
else:
    Pick a player s uniformly at random from S
    T' = T
    S' = S
    for each TA t in T do:
        TA t looks at player s
        if s is t's favorite pick then:
            Match player s to TA t
            T' = T' − {t}
            S' = S' − {s}
            break out of the for loop
    PlayerSorter(T', S')
```

(b) However, the latest draft is right around the corner, and Leah feels the algorithm she designed will not find the purveyor of this Brock Purdy slander fast enough. Help her design an algorithm with a better expected asymptotic runtime bound than the one in part (a). Be sure to explain your algorithm well and prove its correctness. **Formal analysis of runtime is not necessary, but briefly state and explain your improved expected runtime bound.**

If you'd like (for this part only!) you may provide properly-formatted pseudocode to **support** your English explanation of your algorithm. That is, you still **must** provide an English explanation. You can easily format pseudocode using the `verbatim` environment.

(c) Realizing that Leah now has an efficient algorithm to find their identities, the awful TAs, out to disparage Purdy's reputation, further restrict the information they are willing to give her! **Now, each TA either says that the player is their favorite pick or not**, without giving any information on whether their favorite pick is drafted before or after the current player. Does the algorithm in part (a) still work? What about the one in part (b)? Give a brief (1-3 sentence) justification for your answer.

5. [20 pts] Football Ticket Dispute

Thanks to the graciousness of Arvind, the 1210 TAs have been given free tickets to a highly anticipated football game between the 1200 and 1600 TAs. Each ticket is associated with a positive integer ticket number, and Hasit stored the ticket numbers in an n-length array A. Annabella, who dislikes football, is only willing to let the 1210 TAs attend if Hasit can solve her challenge. She created a new n-length array B by changing exactly k numbers in A to 0, and provided a function `arrSum(int low, int high, int[] arr)` that gives the sum of `arr[low ... high]` in $O(1)$ time. Hasit must come up with an efficient, divide and conquer algorithm that takes in A and B and uses Annabella's function to turn B back into A. That is, construct an $O(k \lg n)$ time algorithm that modifies and returns B such that the final state of B equals A.

As an example, if $k = 2$, $A = [25, 33, 19, 14, 9, 38]$, and $B = [0, 33, 19, 0, 9, 38]$, your algorithm should replace the zeros in B with their original values such that $B = [25, 33, 19, 14, 9, 38]$.

(You may assume that $k \ll n$, i.e., k is small enough such that an $O(k \lg n)$ algorithm is asymptotically better than a $\Theta(n)$ algorithm.)

6. [10 pts] Can Gift Cards Be Sentimental?

A fantasy football genius, Arriella has accumulated n unique trophies from past leagues in the form of gift cards (assume n is a power of 2). Each gift card can be classified into one of k different types (Starbucks, Chipotle, Amazon, Raising Canes, etc.), where k is some positive integer.

Deciding that it's time to cash out, Arriella is faced with the difficult task of choosing one of her trophies to liquidate. After painfully reminiscing over each card, Arriella decides that she can only let go of a gift card if it's the most abundant type she has. Specifically, she will only part with a card of type x if more than $n/2$ of her remaining card collection also belongs to type x.

Given that each card in her collection evokes unique memories, Arriella can't identify a card's type by just looking at it. However, she can, in constant time, compare two cards to check if they belong to the same type. **This is the only query she can make.**

Design a $\Theta(n \lg n)$ divide-and-conquer algorithm to find a gift card from Arriella's most common type of fantasy football gift card trophies, assuming that type contains strictly more than $n/2$ cards in the collection. If no type satisfies this condition, the algorithm should return `Nil`.