CIS 1210 — Data Structures and Algorithms
Homework Assignment 4

Assigned: February 07, 2025 **Due:** February 11, 2025

Note: The homework is due **electronically on Gradescope** on February 11, 2025 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. Ed Discussion will also contain a complete sample solution in a pinned post.

D. **Algorithms**: Whenever you present an algorithm, your answer must include 3 separate sections. Please see Ed Discussion for an example complete solution.

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.

1. [20 pts] Valentine's Day Celebration

After weeks of preparing for Valentine's Day, Zimo finds himself at the annual CIS 1210 Valentine's Day fair. The Amy Gutmann Halls are decorated with hearts, streamers, and GSRs are filled with Valentine's envelopes. Tanvi, the head of the Valentine's Day committee, greets him warmly and explains the exciting event.

She tells Zimo about n envelopes arranged in a line in one of the GSRs. Each envelope contains an integer amount of Valentine's grams. The grams inside the envelopes can be positive, negative, or zero: a positive value means Zimo will gain that many grams for free, while a negative value means he will have to give that many grams if he chooses that envelope, and a zero value means he will get no grams at all if he chooses that envelope.

Since Zimo wants to make the most of his time at the fair, he decides to open a consecutive sequence of envelopes to maximize his total Valentine's grams. Help him find the best sequence of consecutive envelopes such that the cumulative product of Valentine's grams is maximized.

Please include an analysis of the Running Time for all parts, but you are only required to give a Proof of Correctness for part (c).

(a) Show that it is possible for Zimo to find the best sequence of consecutive envelopes that includes the first envelope 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 item at the first index.

(b) Give an algorithm for finding the overall best sequence of envelopes in $O(n^2)$ time.

(c) Zimo discovers that Selina, a crafty Valentine's Day Fair enthusiast, has already found a solution to part (b). To help Zimo one up Selina, design a $O(n \lg n)$ divide and conquer algorithm that returns the product of the overall best sequence.

2. [20 pts] Romantic Routes

Gaurav is preparing for Valentine's Day and needs to narrow down his list of romantic locations to only three of them. To make the most of his time, Gaurav wants to find the three locations such that the perimeter of the triangle they form is minimized. Help Gaurav develop an efficient algorithm that, given a set of locations, $N = \{\ell_1, \ell_2, \ldots, \ell_n\}$, $n \geq 3$, with each location ℓ_i having location (x_i, y_i), finds the perimeter δ of a triangle formed by three locations $\{\ell_a, \ell_b, \ell_c\} \subseteq N$ such that

$$\delta = d(\ell_a, \ell_b) + d(\ell_b, \ell_c) + d(\ell_a, \ell_c) \text{ is minimized,}$$

where, for any two locations ℓ_p and ℓ_q, $d(\ell_p, \ell_q) = \sqrt{(x_p - x_q)^2 + (y_p - y_q)^2}$.

3. [10 pts] Vedha the Valentine Prophet

Leah is spending Valentine's Day alone this year, but wants to know when the next time she will spend Valentine's Day with someone else will be. She consults Vedha, an all-knowing fortuneteller, who tells Leah that the next time she will spend Valentine's Day with someone else will be in k years. Vedha does not like to give out too much information, so Leah has to guess how many years into the future this will happen, and Vedha will only tell her if her guess is too early, too late, or correct. Furthermore, Vedha is very busy and limits Leah to only $O(\lg k)$ guesses. Help Leah design an algorithm to correctly guess the value of k in $O(\lg k)$ guesses!

4. [20 pts] Katie Chen's Chocolate Choices

In celebration of Valentine's Day, Katie brings chocolates for all the 1210 TA's. Because she's the best, Katie has brought each TA a box of their favorite chocolates. However, in the frenzy of grading, the chocolates have gotten mixed up and Katie doesn't remember which box belong to which TA. Running out of time, Katie has tasked you with ensuring that each TA receives the correct chocolates! However, you realize that there is no way of knowing which TA prefers which chocolates. The only test you have is to make a TA try a particular chocolate and tell you whether it is too bitter, too sweet, or just right. You may assume that it takes constant time for a TA to test a chocolate. Katie wants to design an efficient algorithm so that each TA will receive their corresponding chocolates in a timely manner.

(a) Consider the following algorithm to figure out which chocolate box is for which TA. Let T denote the group of TAs and C be the set of chocolate boxes. What is the *expected* running time of this algorithm? What is its *worst case* running time? Justify your answer.

```
Match(T, C):
if |T| = |C| = 1 then:
   Assign the chocolates to the TA
else:
   Pick a chocolate box c uniformly at random from C
   for each TA t in T do:
       TA t tries chocolate c
       if it is just right then:
           Assign chocolate c to TA t
           T' = T − {t}
           C' = C − {c}
           break out of the for loop
   Match(T', C')
```

(b) Design an algorithm with a better expected runtime than the one in part (a). Be sure to explain your algorithm well, analyze its expected running time, and analyze its worst case running time.

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

(c) Suppose the following: each time you give a chocolate to a TA, the TA either says that the chocolate is just right or it's not their favorite, without giving any information on whether it is too bitter or too sweet. 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] Jake's Matchmaking

Ja-cupid Fanale is tired of boring Valentine's Days, so he's decided to be the ultimate matchmaker this year! However, between grading assignments and being a student himself, he has a limited amount of time and must decide which dates to set up.

In total, Jake has identified n possible dates (all participants are disjoint) to set up. He measures how long each date would take for him to set up: $s_1, s_2, ..., s_n$, where each amount of time is unique.

Additionally, being the Ja-cupid he is, Jake assigns each date a potential compatibility score between 1 and $2n$. Let $p_1, p_2, ..., p_n$ be the potential compatibility scores such that $\sum_{i=1}^{n} p_i = P$. Note that for every potential date i, there is a corresponding s_i and p_i. Jake wants to find the date taking time s_k satisfying

$$\sum_{s_i < s_k} p_i < \frac{P}{2} \text{ and } \sum_{s_i > s_k} p_i \leq \frac{P}{2}$$

Jake will then make the selected date and all dates taking less time than it happen. For example, if the amounts of time the dates take are 24, 35, 160, 200, 55, 22, and Jake evaluates the compatibilities as 3, 2, 9, 7, 5, 10, respectively, then Jake should select the date with time 55. Note that Jake chooses this date because the total sum of the compatibility scores, P, would be 36, the sum of compatibility scores of dates taking less time than 55 would be $15 < \frac{36}{2}$, and the sum of the scores of all dates taking longer than 55 would be $16 \leq \frac{36}{2}$.

(a) Give an algorithm to compute the date Jake selects in $\Theta(n \log n)$ worst-case time.

(b) Give an algorithm to compute the date in $O(n)$ worst-case time.

6. [10 pts] Shafaa's Sours & Sweets

Shafaa, in preparation for Valentine's Day, has bought a tube of n candy hearts. Inside of this tube, the bottom $\lfloor \frac{n}{2} \rfloor$ candies are Boldly-Bitter Bites while the upper $\lceil \frac{n}{2} \rceil$ candies are Scrumptiously-Satiating Sweets. The candies are labeled $1, 2, ..., n$, where 1 is the lowermost candy and n is the uppermost candy. Candies can only be inserted from the top of the tube and can only be removed from the bottom of the tube; therefore, T acts as a queue.

In order to maximize her enjoyment of these candies during Valentine's Day, Shafaa decides that she wants each Boldly-Bitter Bite to be eaten with a Scrumptiously-Satiating Sweet. To do this, she decides that she wants to reverse all the odd-numbered candies. That is, she wants to swap the first candy with the last odd position candy, swap the third candy with the second to last odd position candy, etc.

In addition to queue T, Shafaa has access to another queue Q and a stack S, both of which are initially empty. Help Shafaa devise an algorithm that properly reorders the candies in T in $O(n)$ time and $O(1)$ extra space *outside of the provided stack and queue, including T*. Your algorithm should return with all elements back in T. Be sure to justify the runtime and space complexity of your algorithm. **For this question, no proof of correctness is required.**