Quiz 1

- Do not open this quiz booklet until directed to do so. Read all the instructions on this page.
- When the quiz begins, write your name on the top of every page of this quiz booklet.
- You have 120 minutes to earn a maximum of 120 points. Do not spend too much time on any one problem. Skim them all first, and attack them in the order that allows you to make the most progress.
- **You are allowed one double-sided letter-sized sheet with your own notes**. No calculators, cell phones, or other programmable or communication devices are permitted.
- Write your solutions in the space provided. Pages will be scanned and separated for grading. If you need more space, write "Continued on S1" (or S2, S3, S4) and continue your solution on the referenced scratch page at the end of the exam.
- Do not waste time and paper rederiving facts that we have studied in lecture, recitation, or problem sets. Simply cite them.
- When writing an algorithm, a **clear** description in English will suffice. Pseudo-code is not required. Be sure to argue that your **algorithm is correct**, and analyze the **asymptotic running time of your algorithm**. Even if your algorithm does not meet a requested bound, you **may** receive partial credit for inefficient solutions that are correct.
- **Pay close attention to the instructions for each problem**. Depending on the problem, partial credit may be awarded for incomplete answers.

Problem	Parts	Points
0: Information	2	2
1: Warmup	4	20
2: Sortid Casino	3	18
3: Restaurant Lineup	1	15
4: Range Pair	2	25
5: Rainy Research	1	20
6: Left Smaller Count	1	20
Total		120

Name: ______________________________

School Email: ______________________________

Problem 0. [2 points] **Information** (2 parts)

(a) [1 point] Write your name and email address on the cover page.

(b) [1 point] Write your name at the top of each page.

Problem 1. [20 points] **Warmup**

(a) [5 points] Given array A of n integers, the Python function below appends all integers from set $\{A[x] \mid 0 \leq i \leq x < j \leq n \text{ and } A[x] < k\}$ to the end of dynamic array B.

```
def filter_below(A, k, i, j, B):
    if (j - i) > 1:
        c = (i + j) // 2
        filter_below(A, k, i, c, B)
        filter_below(A, k, c, j, B)
    elif (j - i) == 1 and A[i] < k:
        B.append(A[i])
```

Argue the **worst-case** running time of `filter_below(A, k, 0, len(A), [])` in terms of $n =$ `len(A)`. You may assume that n is a power of two.

(b) [5 points] The integer array $A = [4, 3, 1, 5, 0, 2]$ is not a heap. It is possible to make A either a max or min heap by swapping two integers. State two such integers.

(c) [5 points] Let T be a binary search tree storing n integer keys in which the key k appears $m > 1$ times. Let p be the lowest common ancestor of all nodes in T which contain key k. Prove that p also contains key k.

(d) [5 points] Given the hash family $\mathcal{H} = \{h_a(k) = a(k + a) \bmod m \mid a \in \{1, \dots, m\}\}$ and some key $k_1 \in \{0, \dots u - 1\}$ where $2 < 2m < u$, find a key $k_2 \in \{0, \dots, u - 1\}$ with $k_2 \neq k_1$ such that $h_a(k_1) = h_a(k_2)$ for every $h_a \in \mathcal{H}$.

Problem 2. [18 points] **Sortid Casino** (3 parts)

Jane Stock is secret agent 006. She is searching for criminal mastermind Dr. Yes who is known to frequent a fancy casino. Help Jane in each of the following scenarios. Note that each scenerio can be **solved independently**.

(a) [6 points] A dealer in the casino has a deck of cards that is missing 3 cards. He will help Jane find Dr. Yes if she helps him determine which cards are missing from his deck. A full deck of cards contains kn cards, where each card has a value (an integer $i \in \{1, \ldots, n\}$) and a suit (one of k known English words), and no two cards have both the same value and the same suit. Describe an efficient[1] algorithm to determine the value and suit of each of the 3 cards missing from the deck.

[1] By "efficient", we mean that faster correct algorithms will receive more points than slower ones.

(b) [6 points] The dealer doesn't know Dr. Yes, but he knows that Dr. Yes is one of the k best players in the casino. Jane scans the room and for each of the $p > k$ players, she transmits back to headquarters a pair (c, ℓ) representing the number of chips c and location ℓ of the player. Assuming that no player has the same number of chips, describe an efficient algorithm for headquarters to determine the locations of the k players in the casino who have the most chips.

(c) [6 points] After determining the locations of the k players with the most chips, Jane observes the game play of each of them. She watches each player play exactly $h < k$ game rounds. In any game round, a player will either win or lose chips. A player's **win ratio** is one plus the number of wins divided by one plus the number of losses during the h observed hands. Given the number of observed wins and losses from each of the k players, describe an efficient algorithm to sort the players by win ratio.

Problem 3. [15 points] **Restaurant Lineup**

Popular restaurant Criminal Seafood does not take reservations, but maintains a wait list where customers who have been on the wait list longer are seated earlier. Sometimes customers decide to eat somewhere else, so the restaurant must remove them from the wait list. Assume each customer has a different name, and no two customers are added to the wait list at the exact same time. Design a database to help Criminal Seafood maintain its wait list supporting the following operations, each in $O(1)$ time. State whether each operation running time is worst-case, amortized, and/or expected.

- `add_name(`x`)`: add name x to the back of the wait list
- `remove_name(`x`)`: remove name x from the wait list
- `seat()`: remove and return the name of the customer from the front of the wait list

Problem 4. [25 points] **Range Pair** (2 parts)

Given array $A = [a_0, a_1, \dots, a_{n-1}]$ containing n **distinct** integers, and a pair of integers (b_1, b_2) with $b_1 \leq b_2$, a **range pair** is a pair of indices (i, j) with $i \neq j$ such that the sum $a_i + a_j$ is within range, i.e., $b_1 \leq a_i + a_j \leq b_2$. Note that parts (a) and (b) can be **solved independently**.

(a) [10 points] Assuming $b_2 - b_1 < 6006$, describe an $O(n)$-time algorithm to return a range pair of A with respect to range (b_1, b_2) if one exists. State whether your algorithm's running time is expected, worst-case, and/or amortized.

(b) [15 points] Assuming $\log_n(\max A - \min A) < 6006$ (with no restriction on b_1 or b_2), describe an $O(n)$-time algorithm to return a range pair of A with respect to range (b_1, b_2) if one exists. State whether your algorithm's running time is expected, worst-case, and/or amortized.

Problem 5. [20 points] **Rainy Research**

Mether Wan is a scientist who studies global rainfall. Mether often receives data measurements from a large set of deployed sensors. Each collected data measurement is a triple of integers (r, ℓ, t), where r is a positive amount of rainfall measured at latitude ℓ at time t. The **peak rainfall** at latitude ℓ **since** time t is the maximum rainfall of any measurement at latitude ℓ measured at a time greater than or equal to t (or zero if no such measurement exists). Describe a database that can store Mether's sensor data and support the following operations, each in worst-case $O(\log n)$ time where n is the number of measurements in the database at the time of the operation.

- `record_data`(r, ℓ, t): add a rainfall measurement r at latitude ℓ at time t
- `peak_rainfall`(ℓ, t): return the peak rainfall at latitude ℓ **since** time t

Problem 6. [20 points] **Left Smaller Count**

Given array $A = [a_0, a_1, \ldots, a_{n-1}]$ containing n **distinct** integers, the **left smaller count array** of A is an array $S = [s_0, s_1, \ldots, s_{n-1}]$ where s_i is the number of integers in A to the left of index i with value less than a_i, specifically:

$$s_i = |\{j \mid 0 \leq j < i \text{ and } a_j < a_i\}|.$$

For example, the left smaller count array of $A = [10, 5, 12, 1, 11]$ is $S = [0, 0, 2, 0, 3]$. Describe an $O(n \log n)$-time algorithm to compute the left smaller count array of an array of n distinct integers. State whether your algorithm's running time is worst-case, amortized, and/or expected.

SCRATCH PAPER 1. DO NOT REMOVE FROM THE EXAM.

You can use this paper to write a longer solution if you run out of space, but be sure to write "Continued on S1" on the problem statement's page.

SCRATCH PAPER 2. DO NOT REMOVE FROM THE EXAM.

You can use this paper to write a longer solution if you run out of space, but be sure to write "Continued on S2" on the problem statement's page.

SCRATCH PAPER 3. DO NOT REMOVE FROM THE EXAM.

You can use this paper to write a longer solution if you run out of space, but be sure to write "Continued on S3" on the problem statement's page.

SCRATCH PAPER 4. DO NOT REMOVE FROM THE EXAM.

You can use this paper to write a longer solution if you run out of space, but be sure to write "Continued on S4" on the problem statement's page.