Quiz 2

- 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 90 minutes to earn a maximum of 90 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 two 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) 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: Red vs. Blue	1	16
2: Color Avoidance	1	16
3: Performance Programming	1	18
4: Toting Tea	1	18
5: Placing Posts	1	20
Total		90

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. [16 points] **Red vs. Blue**

A **red-blue** graph is any dense[1] directed simple[2] graph where:

- every vertex and edge is labeled with a positive integer weight, and
- every vertex is colored either red, blue, or black.

A **red-blue path** is any path in a red-blue graph starting at a red vertex and ending at a blue vertex. The **red-blue weight** of a red-blue path is the sum of the weights of all vertices and edges in the path. Given a red-blue graph G, describe an algorithm to return the minimum red-blue weight of any red-blue path in G, with a running time that is **linear** in the size of G.

[1]Recall a graph $G = (V, E)$ is dense if $|E| = \Omega(|V|^2)$.

[2]Recall a graph is simple if it has no self-loops and no multi-edges.

Problem 2. [16 points] **Color Avoidance**

A graph is **vertex-colored** if each vertex has been assigned a color.

- The **chromatic number** of a path in a vertex-colored graph is the number of distinct colors assigned to vertices on the path.
- A **path-monochromatic** graph is any vertex-colored graph where every pair of vertices assigned the same color is connected by a path with chromatic number 1.

Given an undirected connected simple path-monochromatic graph and two distinct graph vertices, describe an **efficient** algorithm to return the smallest chromatic number of any path between them.

Problem 3. [18 points] **Performance Programming**

Concert cellist Momo Ya plans to hold a concert in a European city every night of December (from Dec. 1st to the 31st), starting with a concert in Vienna on Dec. 1st.

- Between concerts, she will take a single **morning bus** to the city she will perform in next. She may play in the same city twice, but she won't play in the same city two nights in a row.
- Ya has a listing of all b one-way morning bus routes available between the c cities in Europe.
- For each city, she has estimated the **expected revenue** she would earn if she were to perform there. However, if at least one other musician has a concert in the same city on the same date, her expected revenue would be halved.

Given the bus listing, her revenue estimates, and a list of the m concerts (a city and date) that other musicians have already scheduled, describe an **efficient** algorithm to determine a city to perform in each night of December that will maximize Ya's expected performance revenue while on tour.

Problem 4. [18 points] **Toting Tea**

Commodore Jill Swallow[3] wants to find a home port to base her fleet of s ships. Swallow has a map showing the n **ports** in the West Indies and m two-way **shipping lanes** directly connecting pairs of them (assume every port is reachable from every other port via shipping lanes).

- Exactly $s + 1$ of the ports are marked as **client ports**. Upon choosing a home port (which should be a client port), Swallow will assign each ship in her fleet to a different client port to which the ship must make a round-trip tea delivery from her home port each month.
- Each port p is labeled with a **transit fee** f_p: the positive integer number of coins a ship must pay to transfer there from one shipping lane to another. Swallow is very rich, so you can assume her ships will always carry enough coins to cover transit fees.
- Each shipping lane ℓ is labeled with a **congestion number** k_ℓ: the positive integer number of less-well-armed competitors operating on that lane. Whenever one of Swallow's ships shares a shipping lane with a competitor, the competitor will give her ship a "donation" of 100 coins.

Given Swallow's map, describe an $O(nm)$-time algorithm to determine a port to base her fleet so as to minimize her monthly net operating costs (transit fees minus donations), assuming $s = O(\frac{n}{\log n})$.

[3]Swallow is an innocent tea merchant and definitely not a pirate.

Problem 5. [20 points] **Placing Posts**

Misty the Bear, tasked with boredom prevention, wants to build a network of one-way zip-lines to help visitors travel within her local state park.

- There are p **park locations** where posts may be built to support zip-lines.
- Each location ℓ has a known **elevation** $e(\ell)$: an integer number of feet above sea level.
- A **zip-line**, designated by ordered pair (a, b), must start atop a post at location a and end atop of a post at location b. For a zip-line (a, b) to **function**, it must drop by at least 10 feet[4].
- Each location supports one post, and any zip-line connected to a post must connect at its top.

Given a proposed set of z zip-lines and the elevation of each of the p park locations, describe an $O(z + p)$-time algorithm to return a positive integer post height for each location, that will allow every proposed zip-line to function, or return that no such height assignment is possible.

[4] i.e., two posts of heights h_a and h_b must be built at a and b respectively such that $h_a + e(a) \geq h_b + e(b) + 10$.

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.