Problem Set 6

All parts are due on April 5, 2018 at 11PM. Please write your solutions in the LaTeX and Python templates provided. Aim for concise solutions; convoluted and obtuse descriptions might receive low marks, even when they are correct. Solutions should be submitted on the course website, and any code should be submitted for automated checking on `py.mit.edu/6.006`.

Problem 6-1. [30 points] **Graph Practice** A graph $G = (V, E)$ has two common representations. An *adjacency matrix* is a $|V| \times |V|$ matrix of boolean values, where the matrix element in row i and column j is 1 if edge (v_i, v_j) is in E, and 0 otherwise. An *adjacency list* is a $|V|$ length list of lists, where the ith list contains index j if edge (v_i, v_j) is in E.

(a) [8 points] Draw the **directed graph** associated with each of the following graph representations. G_1 is represented by an adjacency matrix, while G_2 is represented as an adjacency list. For consistency, please draw your vertices to lie roughly on a circle ordered clockwise by index.

```
G1 = [[0 1 0 1 0],
      [1 0 0 0 1],
      [1 0 0 0 0],
      [1 0 1 0 1],
      [1 0 0 1 0]]
```

```
G2 = [[1,5],
      [2,3,6],
      [],
      [0,1,2],
      [],
      [0,2,4],
      [2,4,5]]
```

(b) [12 points] Draw two different **connected** undirected graphs on the set of five vertices, $V = \{v_0, v_1, v_2, v_3, v_4\}$. One graph should be a tree and the other should contain exactly one cycle. Write down adjacency matrix **and** an adjacency list representations for each of your graphs.

(c) [10 points] The **separation between two vertices** in an undirected graph is the length of the shortest path between them. The **separation of an undirected graph** is the maximum separation between any two vertices in the graph. The **maximum separation of a vertex** in an undirected graph is the maximum separation between the vertex and any other vertex in the graph. A vertex is a **centroid** of an undirected graph if its maximum separation is minimum across all vertices. Let G be a **connected** undirected graph, with d the maximum separation of G and c the maximum separation of a centroid of G. Prove that $c \leq d \leq 2c$.

Problem 6-2. [30 points] **Consulting**

(a) [5 points] **Security Breach:** SicroMoft, a software company, has experienced a security breach in its company email system. An employee's computer has been infected with a computer virus, that copies itself to another computer if an email is sent to it from the infected computer. Security analysts have identified the computer that was first infected with the virus, and has obtained a log of all emails sent between company computers since the time of first infection. Logged emails are sorted by the time at which the email was sent from one computer to another. Given this log, describe a linear time algorithm to determine which company computers have been infected by the virus.

(b) [5 points] **Gamer Latency:** Four gamers, AceArezu34, BenignBilly56, ClaudiaTheCrusher94, and DastardlyDarius73, regularly engage in intense games of competitive Dandy Drush™. One gamer hosts the game, while the others connect to the host's machine over the internet. For elite gamers, **latency**, or the amount of time it takes for an update to travel from the game's server's to a player's screen, is a big concern! Assume that latency is directly proportional to the number of internet routers the data must pass through to reach its destination. The four gamers decide that the user who hosts their games should be the user who minimizes the sum of latencies of all four users. Note that the host will always have zero latency. Given a map of n network routers from their internet service provider (obtained via questionable means), describe a linear time algorithm to compute who should host their game.

(c) [10 points] **Party Separation:** Haddy Ceron has gotten into trouble with some of her, shall we say, difficult friends: her friends do not get along with each other. She would like to invite them all to a party, but that would cause too much drama. Haddy decides instead to throw two separate parties, one on Friday and one on Saturday night. She wants to invite each of her friends to exactly one of the parties, but not to both. Haddy is specifically worried about conflicting pairs of friends who intensely dislike each other. Given a list of conflicting pairs among her friends, describe a linear time algorithm to determine whether Haddy can assign friends to parties, without assigning any conflicting pair of friends to the same party.

(d) [10 points] **Ground Transportation:** Tim the Beaver needs to get from MIT to a recruiting event in San Francisco, but the airlines no longer allow beavers to fly. Tim cannot drive, so he decides to travel to his destination via bus and train. Tim vastly prefers trains over buses, and wants his trip to be **bus-sparse**: the trip should not contain a transfer from a bus to another bus, only from a bus to a train, a train to a bus, or a train to a train. Given a map of bus and train routes across the country, describe a linear time algorithm to find a bus-sparse itinerary containing the fewest vehicle transfers, or tell Tim that no bus-sparse itinerary exists to his destination.

Problem 6-3. [40 points] **Number Puzzles**

A **15-number puzzle** consists of 15 numbered slide-able square **pieces** placed on a 4×4 grid called a **board**, with one grid square left uncovered. Pieces may slide into the empty grid square, swapping the piece with the empty space; we call such an operation a **move**. The figure below depicts the puzzle in a solved configuration [left] and an unsolved configuration [right]. The goal of the puzzle is to transform an unsolved configuration to a solved configuration by performing a sequence of moves.

15	14	13	12
11	10	9	8
7	6	5	4
3	2	1	0

14	0	13	12
15	11	10	8
7	6	9	4
3	2	5	1

In this problem, we generalize the 15-number puzzle to an $(n \times m - 1)$-number puzzle, containing $n \times m - 1$ numbered pieces in an $n \times m$ board, with a 0 marking the empty square. A move of a piece up, down, left, or right into the empty square will be denoted by the characters '`U`', '`D`', '`L`', and '`R`' respectively. For example, the unsolved configuration above can be solved via the sequence of moves `(R, U, L, L, U, U, L)`.

(a) [2 points] **Configurations:** Compute the number of possible placements of $n \times m - 1$ pieces within an $n \times m$ board.

(b) [3 points] **Implicit Graph:** Given a board configuration C, another board configuration C' is a **neighbor** of C if C can be transformed into C' via a single piece move. What is the minimum and maximum number of neighbors of any $(n \times m - 1)$-puzzle configuration? Describe how to compute all neighbors of a given configuration in $O(nm)$ time.

(c) [10 points] **Shortest Solve:** Given a board configuration, describe an $O((nm + 1)!)$ time algorithm to output a shortest sequence of moves that solves the board, if such a sequence of moves exists.

(d) [25 points] **Implement:** Write the Python function `solve_puzzle(config)` that implements your puzzle solving algorithm. An input configuration will be a length m tuple of length n tuples storing the integers 0 to $n \times m - 1$ in some permutation, e.g., the right board would be `((14,0,13,12),(15,11,10,8),(7,6,9,4),(3,2,5,1))`. The function should return a tuple of moves from the character set {'`U`', '`D`', '`L`', '`R`'} that when applied to the input configuration solves the puzzle, or return `None` if no such sequence exists. Submit your code online at `py.mit.edu/6.006`.