Problem Set 4

This problem set is divided into two parts: Part A problems are theory questions, and Part B problems are programming tasks.

Part A questions are due on Matt's Birthday, **Tuesday, April 6th** at **11:59PM**.

Part B questions are due **Thursday, April 8th** at **11:59PM**.

Solutions should be turned in through the course website in PDF form using LaTeX or scanned handwritten solutions.

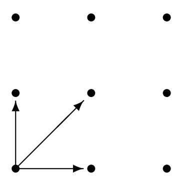
A template for writing up solutions in LaTeX is available on the course website.

Remember, your goal is to communicate. Full credit will be given only to the correct solution which is described clearly. Convoluted and obtuse descriptions might receive low marks, even when they are correct. Also, aim for concise solutions, as it will save you time spent on write-ups, and also help you conceptualize the key idea of the problem.

Part A: Due Tuesday, April 6th

1. **(20 points)** Reasoning about Search

   For each of the following statements, prove the statement or give a *small* counter example to show that it is false. You may use LaTeX to draw example graphs if necessary (the solution template contains a drawing of the following graph to get you started).



   (a) **(6 points)** A cyclic directed graph G has exactly one back edge produced by DFS. It is possible to remove one edge from G to make G acyclic.

   (b) **(6 points)** If a **directed** graph G has some vertices u and v in the same DFS tree and the DFS finishing time for v is after that of u then v is an ancestor of u in the DFS tree.

   (c) **(8 points)** A graph G with n vertices can have DFS produce a tree with depth $n-1$ while BFS on G will have depth of only 1. A root node is defined as having depth 0. (Note: if an example is difficult to draw with LaTeX vectors you may use text to describe it)

2. **(14 points)** Bipartite graphs

   An undirected graph is called *bipartite* if its nodes can each be assigned a color, either red or blue, such that no red node is adjacent to another red node, and no blue node is adjacent to another blue node. Give an efficient algorithm to determine if a graph is bipartite. What is its running time?

   Hint: You may want to relate bipartiteness of a graph to the presence/absence of odd length cycles in it.

3. **(16 points)** Flight Planning

   You are traveling an island nation with N cities. You start at city 1 at time 0 and need to get to city N no later than time T. There are M flights each of which flies from some city to some other city. All flights leave at an integer hour and arrive at an integer hour. That is, each flight i leaves some city a_i at an integer time t_{il} and arrives at another city b_i at integer time $t_{ia} > t_{il}$. Assume that if you arrive in a city at time t and there is a flight leaving at time t then you can make the transfer.

   Given the times of the M flights, propose an algorithm based on BFS for determining whether there is a way to get from city 1 to city N in no more than time T. Your algorithm should run in $O(N \cdot T + M)$ time.

Part B: Due Thursday, April 8th

(50 points) Eight Puzzle

The eight puzzle is a game in which 8 tiles numbered 1 through 8 are placed on a $3 * 3$ board. One of the squares of the board always remains uncovered. A possible configuration is the following:

2	6	4
1	7	
8	3	5

In each step the player can slide one of the (at most four) tiles adjacent to the empty square into the empty square. The above configuration can for instance be turned into one of the following configurations:

2	6	4
1		7
8	3	5

2	6	
1	7	4
8	3	5

2	6	4
1	7	5
8	3	

The goal of the game is to turn the initial configuration into the following configuration:

1	2	3
4	5	6
7	8	

Note that that there are $9! = 362,880$ possible different configurations of the board. Not every of them needs be reachable from every other one.
Your task in this problem is to write a program that reads a description of the problem from an input file ps4b.in, and writes a solution to an output file ps4b.out.

Input Format

Lines 1…3: The lines describe initial configuration. The number 0 denotes the blank square.

Sample Input

```
4 1 3
2 6 0
7 5 8
```

Output Format

Line 1: The minimum number of steps to solve the puzzle in the input. If the puzzle is not solvable, just output "`unsolvable`"

Sample Output

```
7
```

Note: If your algorithms seems to take too long to run, you should rethink your algorithm. Even a poor implementation should only take around 10 seconds on a standard Athena Cluster computer or dialup.