CS 540: Introduction to Artificial Intelligence
Homework Assignment # 2

Assigned: 2/12
Due: 2/19 before class

Hand in your homework:

If a homework has programming questions, please hand in the Java program. If a homework has written questions, please hand in a PDF file and name it hwX.pdf where X is the homework number. Regardless, please zip all your files into hwX.zip where X is the homework number. Go to UW Canvas, choose your CS540 course, choose Assignment, click on Homework X: this is where you submit your file.

Late Policy:

All assignments are due at the beginning of class on the due date. One (1) day late, defined as a 24-hour period from the deadline (weekday or weekend), will result in 10% of the total points for the assignment deducted. So, for example, if a 100-point assignment is due on a Wednesday 9:30 a.m., and it is handed in between Wednesday 9:30 a.m. and Thursday 9:30 a.m., 10 points will be deducted. Two (2) days late, 25% off; three (3) days late, 50% off. No homework can be turned in more than three (3) days late. Written questions and program submission have the same deadline.

Assignment grading questions must be raised with the instructor within one week after the assignment is returned.

Collaboration Policy:

You are to complete this assignment individually. However, you are encouraged to discuss the general algorithms and ideas with classmates, TAs, and instructor in order to help you answer the questions. You are also welcome to give each other examples that are not on the assignment in order to demonstrate how to solve problems. But we require you to:

- not explicitly tell each other the answers
- not to copy answers or code fragments from anyone or anywhere
- not to allow your answers to be copied
- not to get any code on the Web

In those cases where you work with one or more other people on the general discussion of the assignment and surrounding topics, we suggest that you specifically record on the assignment the names of the people you were in discussion with.

Question 1: Successor Function [50 points]

This is a programming question. The solution to the programming problem should be coded in Java, and **you are required to use only built-in libraries** to complete this homework. Please submit a source code file named **successor.java** with **no package statements**, and make sure your program is runnable from command line on a department Linux machine. We provide a skeleton successor.java code that you can optionally use, or you can write your own.

The goal of this assignment is to become familiar with the state space in a real-world problem.
You are given three water jugs with capacity A, B, C liters, respectively. A, B, C are positive integers. At each step, you can perform one of these actions:

- Empty a jug
- Fill a jug
- Pour water from one jug to another until either the former is empty or the latter is full.

Write a program **successor.java** to print the successor states of an input state.

- Input: 6 integers A, B, C, a, b, c, where (A, B, C) are the capacity of the jugs, and (a, b, c) are the current amount of water in each jug. You may assume that the input is valid, namely $a \leq A, b \leq B, c \leq C$, and A, B, C are positive integers, and a, b, c are non-negative integers.
- Output: Print the list of successor states reachable in one step from (a, b, c), one one each line. The lines do not need to be sorted. The successors should not include (a, b, c) itself.

Here are some examples of running the program from the command line. The inputs and outputs are space-separated with no comma. Please follow the same input/output format.
Example 1:

```
$java successor 3 2 1 3 2 0
0 2 0
3 0 0
3 2 1
2 2 1
3 1 1
```

Example 2:

```
$java successor 11 5 2 6 3 1
0 3 1
6 0 1
6 3 0
11 3 1
6 5 1
6 3 2
9 0 1
7 3 0
4 5 1
6 4 0
5 3 2
6 2 2
```

Question 2: State Space [50 points]

An (m, n, k)-puzzle is a sliding puzzle with m columns, n rows, and k empty squares, where $1 \leq k < mn$. As usual, one move is to move a single tile to an adjacent (up, down, left, right) empty square, if available.

1. (10 points) How many tiles are there in a (m, n, k)-puzzle in general? For example, there are 8 tiles in a $(3, 3, 1)$-puzzle.

2. (20 points) Let each tile have a distinct name. How many distinct states are there in the state space? Show your derivation.

3. (20 points) Draw a graph that corresponds to the state space of a $(2, 2, 1)$-puzzle, and **briefly describe your graph**. This is not an art project: You may represent the states in ways easy for you to type, and you do not necessarily need drawing programs – you may even use plaintext. You can also hand draw the graph, but please clearly show the nodes and edges.