

# CS 461 – ARTIFICIAL INTELLIGENCE

## HOMEWORK #1 (5%)

Assigned: Mon 15 Oct 2018

Due: Mon 22 Oct 2018 \*\* 2 pm \*\*

You can do this homework in groups of five or less. Your group for this homework may simply coincide with your term project group. In any case, do not forget to indicate clearly the individuals who are submitting this homework (i.e., write at most five names).

You must email your entire homework (just a single submission per group) to our TA (Caner Mercan, PhD student in CS Department). Any programming language can be used as long as you yourself have installed the programming environment (interpreter or compiler) for it.

Our TA will soon send you a brief note explaining the mechanics of submissions. He may or may not ask you to also submit hard copy.

Any of the group members should be prepared to give a demo (individually and using his/her own notebook) when requested to do so by the TA.

---

### PROBLEM

Consider the following puzzle:

*You have two jugs. One holds 10 liters of water when completely full, the other holds 6 liters when completely full. You have an unlimited source of water to fill either or both jugs, and you may do so as often as you like, but there are no markings on the jugs, so if you fill a jug the only way to know how much water you've put in is if you fill it to the top. In addition, you have amazing accuracy in pouring water from one jug to another and never lose a drop. When you do choose to pour water from one jug to another you must keep pouring until either the receiving jug is full or the source jug is emptied of water. You can also always choose to fully empty either jug. Can you get 8 liters of water in the larger jug?*

a. Formulate this puzzle as a state space search. (This part -- that is, part a -- is for you to ponder about. Your answers should appear as comments in the beginning of your code.)

- What are the states?
- What are the initial and goal states?
- What are the operators?
- What is the branching factor?

b. Run your search program (see below) 5 times and let it list a path from the initial state to the goal state for each run.

You must use **Nondeterministic Search**. (Just implement, in a straightforward manner, the pseudocode given in Winston, chap. 4.) You must check for repeated states.

Your program should have a simple control for 'single stepping' so that you and the TA can inspect the intermediate stages of the problem solving process in an incremental fashion. Needless to say, this is also useful for debugging your program during the development stage.

**N.B. IF OUR TA ASKS YOU TO SUBMIT HARD COPY, MAKE SURE THAT ANYTHING YOU SUBMIT IS MACHINE-GENERATED. THUS, IF THERE IS A HANDWRITTEN ADDITION OR CORRECTION ON A PRINTOUT, YOU'LL DEFINITELY LOSE POINTS. THIS INCLUDES EVEN THE NAMES OF GROUP MEMBERS.**

**LATE POLICY:** Late submissions will first have 1% deduced categorically. Then they'll have 1% deduced for every late day.