**The** **Jugs** problem

# In the movie "[Die Hard 3](https://www.youtube.com/watch?v=BVtQNK_ZUJg)", liam Neeson and Laurence Fishburne were confronted with a water jug puzzle; given a 3-gallon jug and a 5-gallon jug, they were asked to fill the 5-gallon jug with exactly 4 gallons, or everyone dies!

# In this assignment, we will try to solve a generalize version of this puzzle, where we intend to fill the water jug A with any amount of water, given any jugs size. You are given two jugs, A and B, and an infinite supply of water (Obviously, you are not in California any more). You can take one of the following 3 actions at any point: (1) fill a jug, (2) empty a jug, and (3) pour water from one jug to the other. Pouring from one jug to the other stops when either the first jug is empty or the second jug is full, whichever comes first. For example, if A has 5 gallons and B has 6 gallons and a capacity of 8, then pouring from A to B leaves B full and 3 gallons in A.

**Functions in supporting roles (10 points each)**

Define six procedures that perform the basic operations of the two jugs, as described below:

* Fill-A – fill Jug A,

ex., (Fill-A ‘(0 2 5 3)) 🡪 (5 2 5 3)

* Fill-B
* Empty-A

Ex., (Empty-A ‘(3 2 5 3)) 🡪 (0 2 5 3)

* Empty-B
* A-to-B – pour everything from A to B to fill B, whichever comes first, ex.,

(A-to-B '(5 2 5 3)) 🡪 (4 3 5 3) ;; A is not necessarily empty after this action

(A-to-B '(2 0 5 3)) 🡪 (0 2 5 3) ;; B is not necessarily full

* B-to-A

**Be sure to name your function exactly as specified, so as to make TA’s job a little easier.**

**Search (40 points + 10 bonus points)**

Write scheme functions that uses breadth-first search to solve the jugs problem, similar to the solution discussed in class for like problems such as FWGC. The main procedure should be named ***Jugs***. Jugs will accept 3 arguments: Size-A, Size-B, Goal, where Size-A and Size-B are the capacities of the jugs A and B, respectively, and Goal is the desired amount of water in A. You may assume the goal is less than that the capacity of jug A, thus a solution is theoretically possible.

When a solution is found, Jugs should return a list of states, where each state takes the form (Water-in-A Water-in-B Size-A Size-B). It returns '() if a solution can not be found. For example,

(Jugs 5 3 4) 🡪 ((0 0 5 3) (5 0 5 3) (2 3 5 3) (2 0 5 3) (0 2 5 3) (5 2 5 3) (4 3 5 3))

For 10 bonus points, you program should also print the sequence of operation in your solution. For example, the printed [solution](http://www.youtube.com/watch?v=lZ64IR2bz5o) for (Jugs 5 3 4) would be:

6 steps:

Fill-A

A-to-B

Empty-B

A-to-B

Fill-A

A-to-B

Name your program file “*YourLastName2.rkt*” and place it in your web drop box by class time on due date.