

## Problem 1: Blocks world

### This assignment should be done individually

- To do so, create a mooshak account with your ualg login (without @ualg.pt) in mooshak. Example: the student with the number 123456 uses a123456. NB: do not use your name in the login.

- Submit your code to mooshak <http://deei-mooshak.ualg.pt/~jvo/> Problem B up to:

November 4, 2020 – 17h

- A submission will remain *pending* until validated by the Instructor during the lab class. Only *final* submissions will be considered for evaluation. Deadline for validation:

November 17, 2020 – 15h30

### Problem

The Blocks World is a classic AI toy problem. It consists of one dimensional table and a set of two-dimensional blocks where an agent is supposed to setup a given desired configuration (i.e., one or more vertical stacks of blocks) starting from an initial configuration following a few simple rules. See Fig. 1 for an illustration.

In our version of the problem the rules are: i) only one block can be moved at a time: it may either be placed on the table or placed on the top of another block; ii) The only blocks that can be moved are those that do not have any other block(s) on their top.

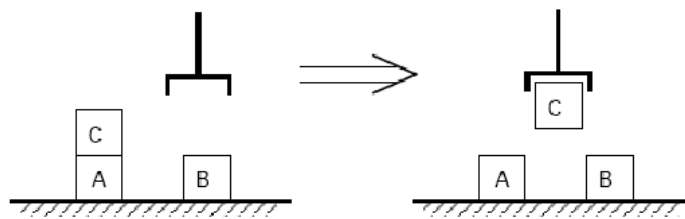


Fig. 1 – An example of the classic Blocks World as shown in [1].

### Task

There are many approaches to address this problem. However, your task is to write a program that, based on the approach followed in Tutorial 1, returns the successive configurations required for achieving the final configuration starting from the initial one, along the shortest path, i.e., along the path with the minimum number of movements.

At this time, only three blocks are considered and for comparing configurations it does not matter the ordering of the stacks on the table. For example, a configuration with the blocks A, B and C on the table is viewed as similar to the configuration with the blocks B, A, and C on the table.

*Please note that any other approach, however meritorious it may be, will be quoted with 0 (zero).*

### **Input**

The input has two lines. The first line represents the initial configuration while the second line denotes the goal configuration. In both lines, the blocks are labelled A, B, and C; a vertical stack is denoted by a sequence of blocks; different stacks are separated by spaces. See the representation of Fig. 1 in example 1.

All given instances will have a (unique) solution.

### **Output**

The sequence of configurations from the initial to the final one (both *inclusivé*), with a blank line separating each one of them. At the end of this sequence, and after a new empty line, a non negative integer with the length of the shortest path found should be also presented.

A configuration will represent each stack of blocks in a separated row. Each stack is within [] and two consecutive blocks within the same stack are separated by a comma followed by a space.

### **Sample Input 1**

AC B  
A C B

### **Sample output 1**

[A, C]  
[B]

[A]  
[B]  
[C]

1

**Sample input 2**

ABC  
A B C

**Sample output 2**

[A, B, C]

[A, B]  
[C]

[A]  
[C]  
[B]

2

**Sample input 3**

CAB  
ABC

**Sample output 3**

[C, A, B]

[C, A]  
[B]

[C]  
[B]  
[A]

[C]  
[A, B]

[A, B, C]

4

**References**

[1] <https://rodneybrooks.com/forai-the-origins-of-artificial-intelligence/>