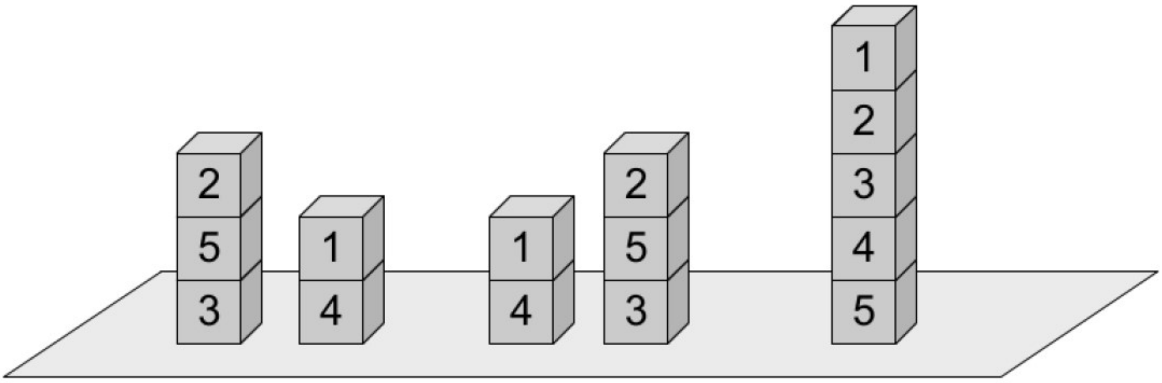
## Week 4

The block's world is a well-known toy domain used in AI for planning problems related to robots. It consists of a set of blocks of various size, shape, and color, possibly identified by a letter or by a number, and placed on a surface (e.g., on a table). The blocks can be stacked into one or more piles, possibly with some constraints (depending, e.g., on their size). The goal is to form a target set of piles starting from a given initial configuration by moving one block at a time, either to the table or atop another pile of blocks, and only if it is not under another block.

In a simple version of this problem, there are five blocks with identical shape and size, numbered from 1 to 5. The figure below shows two possible configurations of such blocks; note that the relative position of the piles does not matter, thus the configuration on the left is the same as the one in the middle. Every block can be either on the table (there are no constraints on the number of piles) or atop any another block. The goal is to stack the blocks in a single pile, in the order shown in the rightmost configuration in the figure, starting from any given set of piles, by moving the smallest possible number of blocks. Only blocks at the top of the current piles can be moved, and can be placed only on the table or atop another pile.




(a) Formulate the above version of the block's world as a search problem, by precisely defining the state space, giving the initial state, the goal test, the set of possible actions and the path cost.

(b) Assume that the initial configuration is the one shown on the left in the figure above, and consider a heuristic function defined as the number of blocks that are not in the highest pile (or in one of the highest piles). Is this heuristic admissible?

(c) In the context of (b), explain why the hill climbing algorithm runs into trouble.

(d) Define another possible admissible heuristic, how would you prove its admissibility?

(e) Given the following search space below, perform Uniform Cost Search starting from node S to find a path from the node to a goal state, ie., a state whose label starts with G.

(f) Repeat the same search with hill climbing

If the agenda has any ties between nodes, i.e. unsure of which selection, then priority should be given to higher in the alphabet. So J has a higher priority than G, and G is higher than B. The numbers on the nodes represent heuristic values.