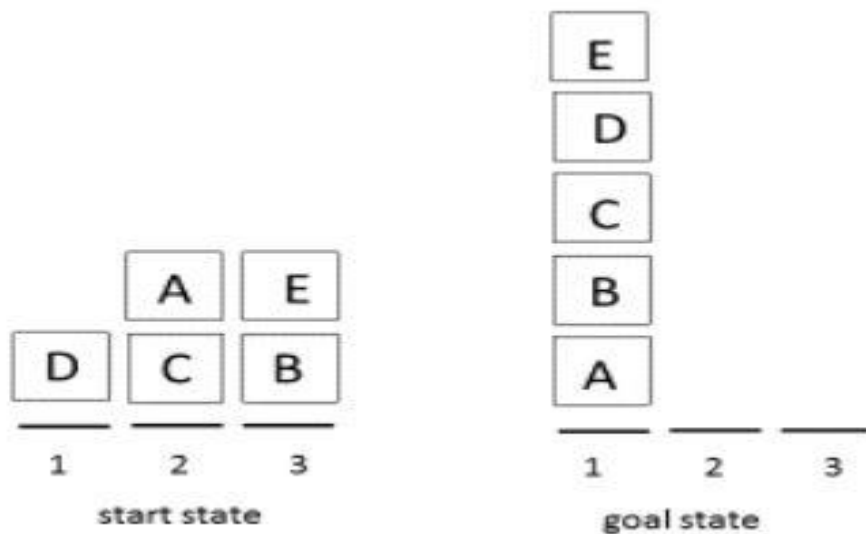


## Problem

The overall goal of this assignment is to implement A\* search and use it to solve a classic AI search problem: the "Blocksworld". This problem involves stacking blocks into a target configuration. You are to write a program to solve random instances of this problem using your own implementation of A\* search. The focus of the project, however, is to use your intuition to develop a heuristic that will make solving various instances of this problem more efficient (solve them in fewer iterations).

Follow the description of A\* in the textbook, i.e. GraphSearch with the frontier represented as a PriorityQueue. (Your code for A\* should be written from scratch by you; however, you may use an existing implementation of PriorityQueue from a library, such as in the STL.) Specifically, you will keep the queue sorted based on  $f(n)=g(n)+h(n)$ .  $g(n)$  is just the path length from the root to the current node. **The main focus of this project will be on developing a heuristic function,  $h(n)$** , for this domain that estimates the distance to the goal (number of moves to solve the problem) and testing how it affects the efficiency of the search -something more sophisticated than just "number of blocks out of place".



examples random state:

```
1 | D
2 | C A
3 | B E
```

```
goal state:
1 | A B C D E
2 |
3 |
```

By experimenting with your program on random problems with different numbers of blocks and stacks, you should generate a table with some Measurements showing how your heuristic performs on different problems.