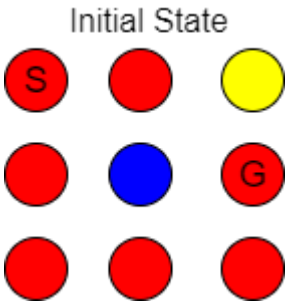
informed search (A*)

Design input & Design state representation:

We can represent the state as a 2D list that contains (N * M) elements, each element has a specific color and position which represents a node or a cell in the board.

For example, the initial state of the board in a problem may be as follows:

[[1, 1, red], [1, 2, red], [1, 3, yellow],
[2, 1, red], [2, 2, blue], [2, 3, red],
[3, 1, red], [3, 2, red], [3, 3, red]]



Design moves:

```
% Movement rules within the grid.
% All moves in this problem have equal cost (0)
move(State, Next, 0, N, M):-
    left(State, Next,N);
    right(State, Next,N);
    up(State, Next,M);
    down(State, Next,M).

left([X,Y,_], [X1,Y1,_], M):-
    Y1 is Y-1, X1 is X, Y1 > 0, Y1 =< M.

right([X,Y,_], [X1,Y1,_], M):-
    Y1 is Y+1, X1 is X, Y1 > 0, Y1 =< M.

up([X,Y,_], [X1,Y1,_], N):-
    X1 is X-1, Y1 is Y, X1 > 0, X1 =< N.

down([X,Y,_], [X1,Y1,_], N):-
    X1 is X+1, Y1 is Y, X1 > 0, X1 =< N.
```

Design heuristic predicate:

```
% Heuristic function to calculate
% the Manhattan distance between two nodes.
calculateH([], [], 0):- !.
calculateH([X1,Y1,_], [X2,Y2,_], H):-
    H is abs(X1-X2) + abs(Y1-Y2).
```

f(n) = g(n) + h(n)

g(n) = exact cost to reach the node n

h(n) = estimated cost from node n to goal

f(n) = estimated total cost of path through n to goal

Design output:

The goal to find path in the board using **Best-First Search** and print at least one path with best cost including its color if more than one exists or no path exist if there are no paths can reach the goal in the board.
As in the example, the goal is:

[1, 1, red] => [2, 1, red] => [3, 1, red] => [3, 2, red] => [3, 3, red] => [2, 3, red]