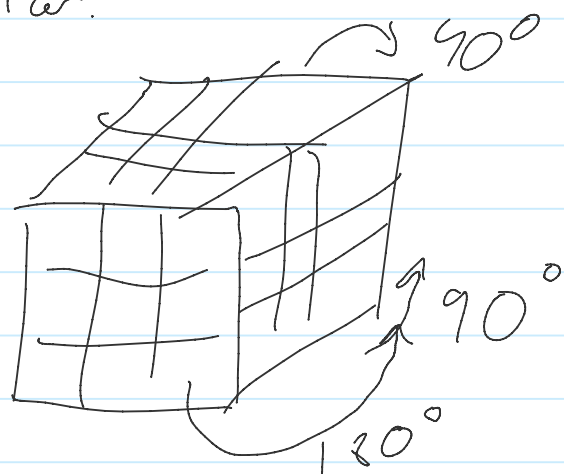


Rubix Cube

3 by 3 by 3 cube
 each cublet face has 1 of 6
 colors

Objective each face has one
 color.



12 possible moves

Dijkstra (sp)

Other example

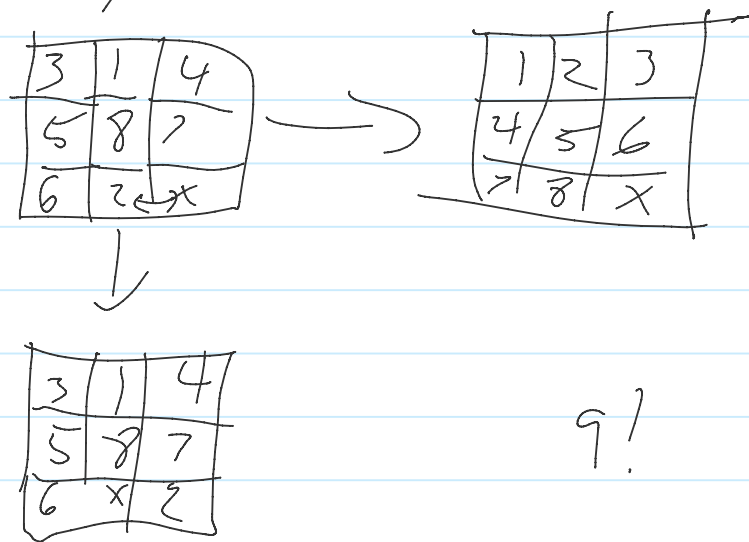
8 puzzle

3x3 grid

exactly 1 empty cell

... 1 ... 1 ... 1 ... 1 ... 1 ... 1 ... 1

each other cell has number 1 through 8 exactly once



Moves consist of swapping blank cells with adjacent cells.

BFS(scrambled)

int dist[# possible puzzles] ← fill ∞

boolean seen[# possible puzzles]; ← filled false

queue q;
 q.add(scrambled);
 seen(scrambled) = true;
 dist(scrambled) = 0;

while (q ≠ ∅) {

cur state = q.poll();

for (all visitable states from cur state)

next state

if (!seen(next state)) {

seen(next state) = true;

dist(next state) = dist(cur state) + 1;

end state break;

if (dist(end state) ≠ ∞) {

print distance

else {

print "not visible"

}

