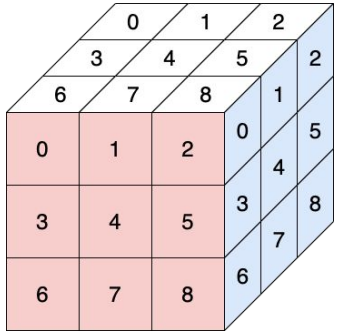
Rubik’s Cube Solver

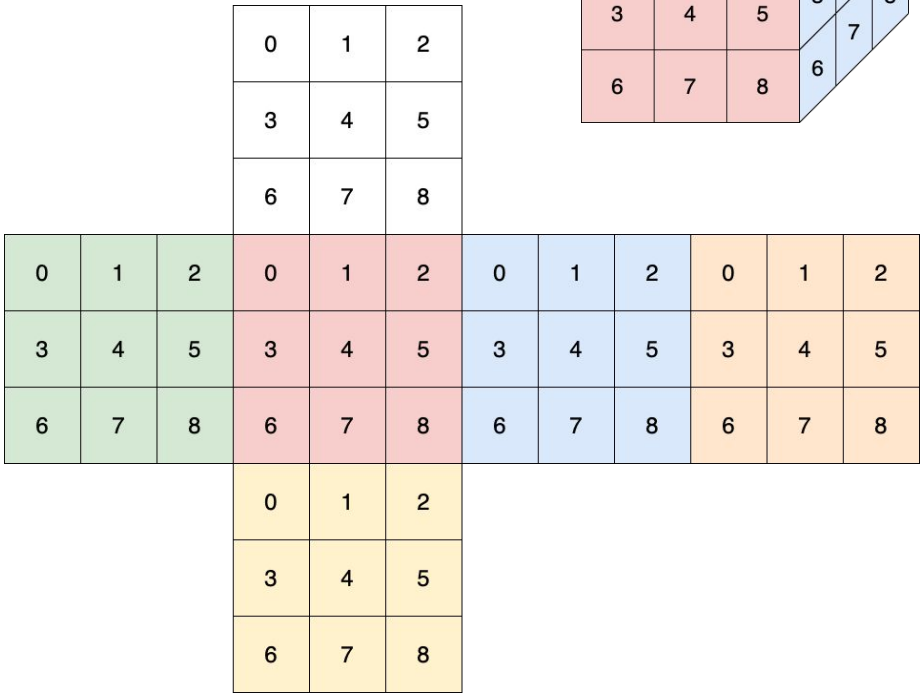
By: Andrew Pobrica & Alex Clarke

Representation

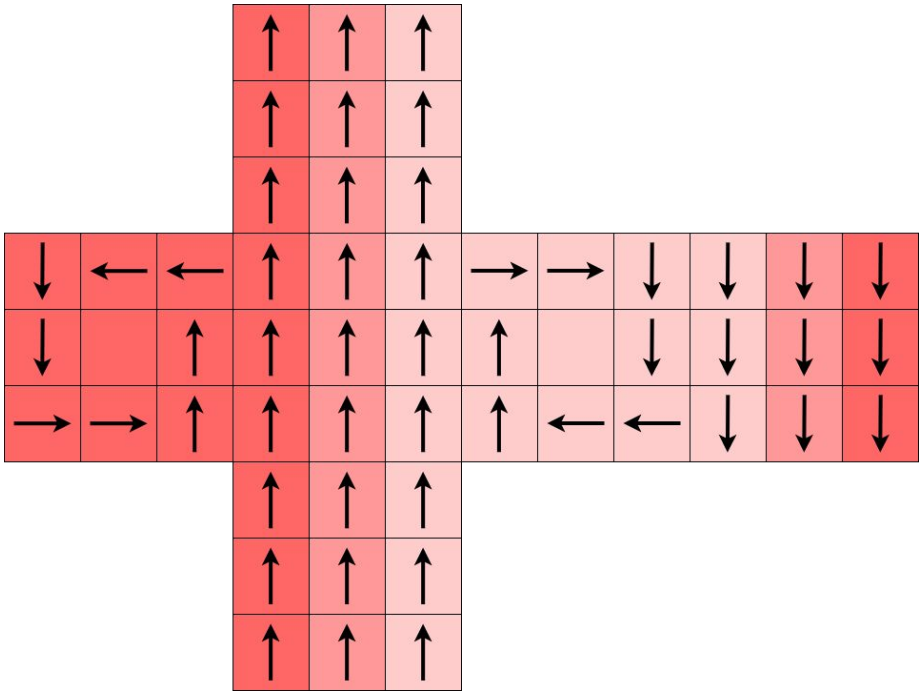
```c
#define SIDES 6
#define N 3

// Color definition.
typedef enum {WHITE, GREEN, RED, BLUE, ORANGE, YELLOW} color_t;

// Side definition.
typedef enum {TOP, LEFT, FRONT, RIGHT, BACK, BOTTOM} side_t;

// N x N x 6 rubix cube definition definition.
typedef color_t cube_t[SIDES][N][N];
```

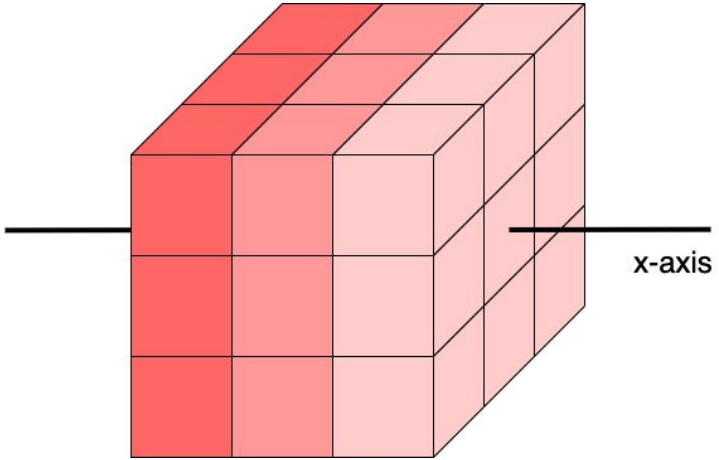
TOP	0	1	2	3	4	5	6	7	8
LEFT	0	1	2	3	4	5	6	7	8
FRONT	0	1	2	3	4	5	6	7	8
RIGHT	0	1	2	3	4	5	6	7	8
BACK	0	1	2	3	4	5	6	7	8
BOTTOM	0	1	2	3	4	5	6	7	8



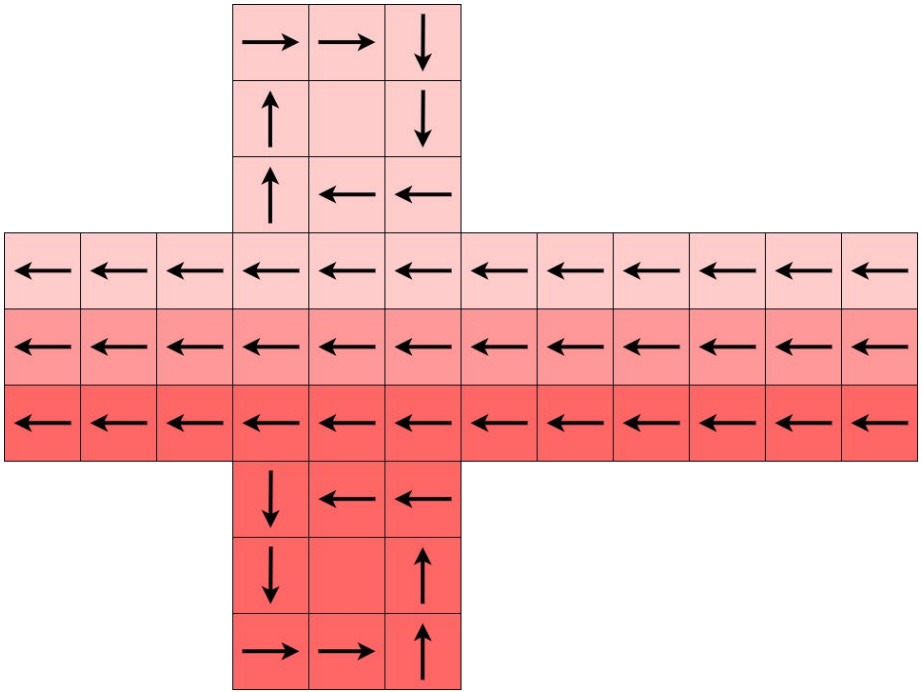


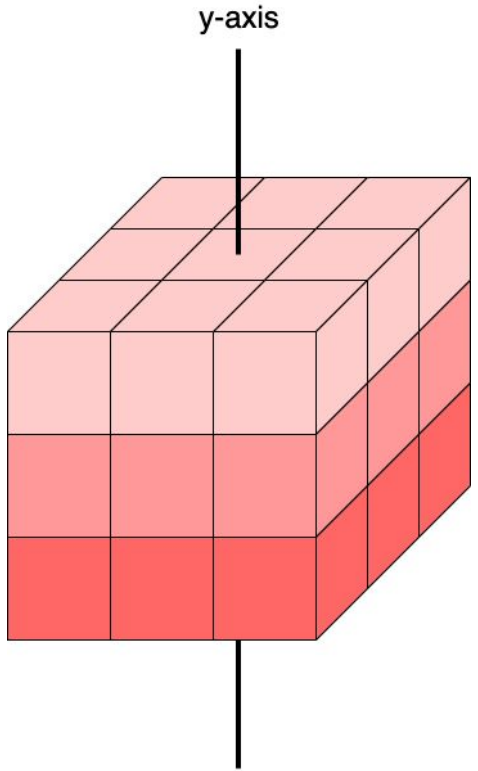
Rotate X



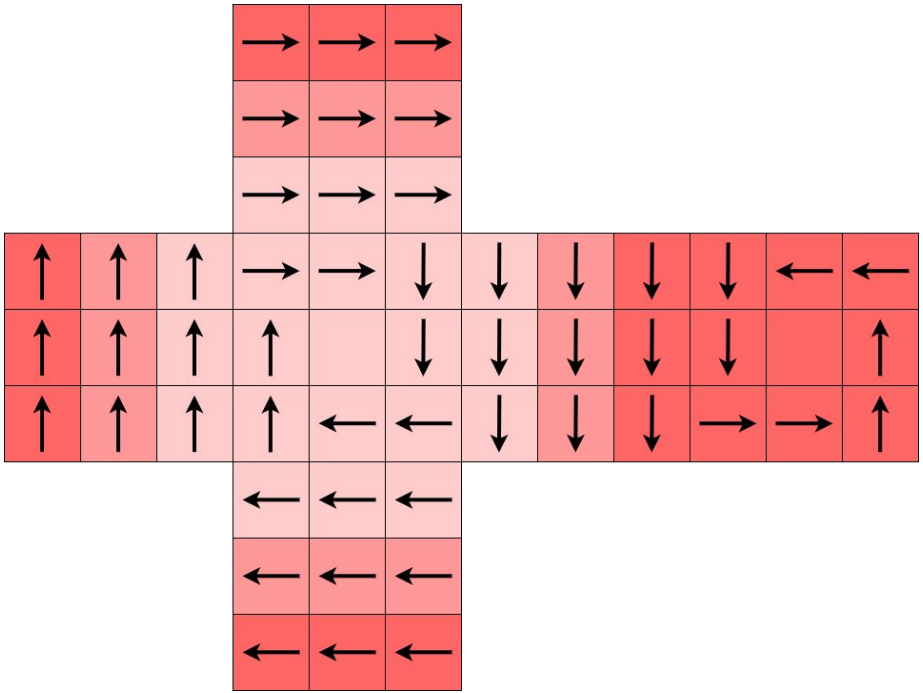


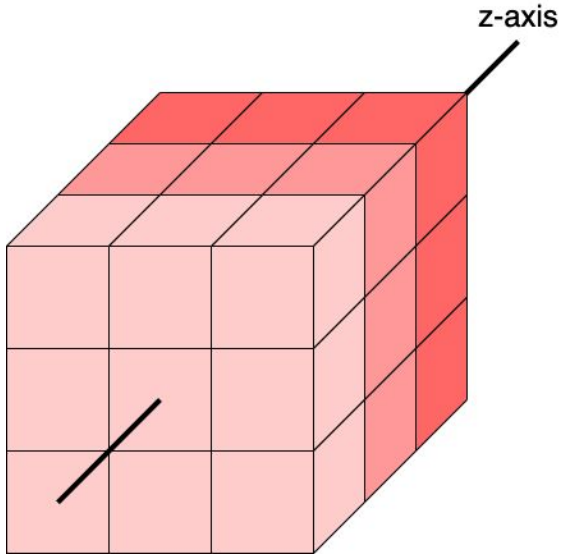
Rotate Y





Rotate Z





Problems

- Special state for edges and corners
- For a traditional 3x3 cube there are 43,252,003,274,489,856,000 possible configurations
- But there’s Gods number
- Max 20 moves to solve any combination on cube
- 1,802,166,800,000,000,000 combinations now
- 4% of all possible combinations [1]

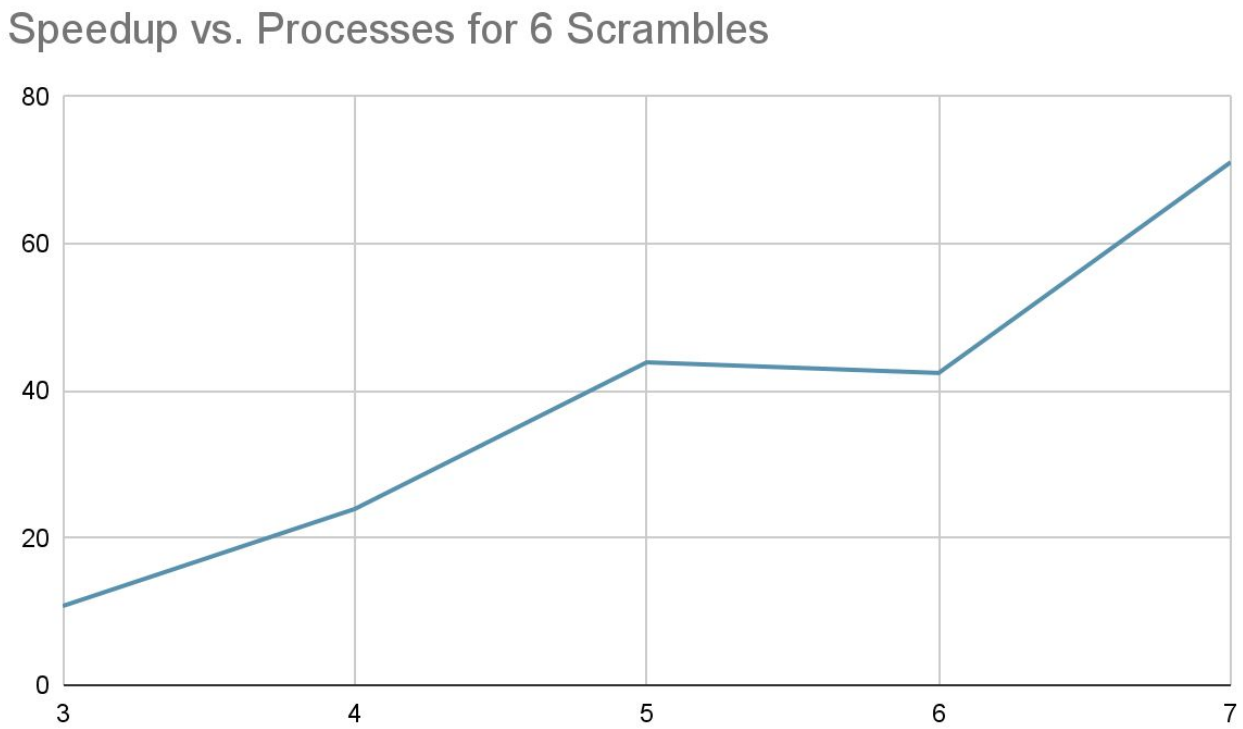
Sequential Solver

- Recursive DFS
- Global array used to store data
- Index into array determined by depth variable incremented at recursive call
- Whenever we find a new better solution copy our current solution array into best array and set best solution to current depth
- First layer returns a struct with best solution and its length

Parallel Solver

- Also Recursive DFS
- Index into array determined by depth variable incremented at recursive call
- Whenever we find a new better solution we malloc an array of our solution size and return it
- Each layer adds its part to solution and returns it
- Still have best solution variable
- First layer returns a struct with best solution and its length
- Process 0 selects best returned solution

Results



Future work

1. Backtracking/loop prevention.
2. Command line arguments.
3. Curses visualization.
4. Non-square combination puzzles.
5. Non-recursive solution for both single core and parallelized
6. Dynamic parallelism with smaller jobs

Citations

1. https://ruwix.com/the-rubiks-cube/gods-number/