TILT BOARD GAME

Team (87)

***Global variables :***

short boardSize; size of any row or column

pair<short, short> target;

vector<vector<char> > gameBoard; saving game board from file

vector<pair<short, short> > currentState;

vector<vector<char> > gameBoardWithoutSLiders; like gameboard but without slider to build every state on it in output

bool reachTarget; check before moving again

vector<pair<short, short> > startState; to start building output from it

vector<pair<short, short> > finalState; to start getting path from it when building output

int queueIndex;

int queueSize;

vector<pair<short, short> > queue[5000000]; for BFS

map<vector<pair<short, short> >, pair<vector<pair<short, short> >, char> > parentState;

to save every statement to avoid redundant with his parent to build the output

int tmpStateIndex;

vector<pair<short, short> > tmpState; to move sliders for each row or column

int minimumNumberOfMoves;

string path;

int slidersCount;

int newStateIndex;

vector<pair<short, short> > newState;

A screenshot of a computer program

Description automatically generated

// **O(N2)**

A computer screen shot of a code

Description automatically generated

// O(K) where k is the number of sliders

A computer screen shot of a code

Description automatically generated

// O(K) where k is the number of sliders

A computer code with text

Description automatically generated with medium confidence

// O(K log(k)) where k is the number of sliders

A screenshot of a computer program

Description automatically generated

// **O(N2)**

A screenshot of a computer program

Description automatically generated

// **O(N2)**

A screenshot of a computer program

Description automatically generated

// **O(N2)**

A screenshot of a computer program

Description automatically generated

// **O(N2)**

A screenshot of a computer program

Description automatically generated

**In case of solvable:** **bounded by O(N2 r2)**, r (=4) is the max number of successors for a board state.

**In case of Not solvable:** **bounded by** , b, s are the number of fixed obstacles and movable sliders, respectively

A screenshot of a computer code

Description automatically generated

// **O(N2)**

A screenshot of a computer

Description automatically generated

// **O(N2)**

\A screenshot of a computer code

Description automatically generated

// O(1)

\A screenshot of a computer code

Description automatically generated

**Sample:** should be **bounded by O(N2 x k)**

**Complete:** should be **bounded by O(k)**

**But in our project we print every statement to target even in complete so its bounded by O(N2 x k)**

Device name: Mac Pro M1 apple chip

1 Small:

Case 1: must be 90 sec

A black background with white text

Description automatically generated

failed

Case 2: must be 60 sec

A black background with white text

Description automatically generated

pass

Case 3: must be 20

A black background with white text

Description automatically generated

pass

2 Medium:

Case 1: must be 70 sec

A black background with white text

Description automatically generated

pass

Case 2: must be 20 sec

A black background with white text

Description automatically generated

pass

Case 3: must be 130 sec

A black background with white text

Description automatically generated

pass

3 Large:

Case 1: must be 70 sec

A black background with white text

Description automatically generated

pass

Case 2: must be 30 sec

A black background with white text

Description automatically generated

pass

Case 3: must be 120 sec

A black background with white text

Description automatically generated

pass