



N-puzzle

Підготував: Пащук Назар Михайлович



Реалізація

Перевірка на розв'язуваність

```
% solvable(+Puzzle) - true, якщо головоломку можна вирішити
solvable(Puzzle) :-
    sum_row_inversions(Puzzle, R),
    empty_square_row(Puzzle, I),
    0 is (R + I) mod 2.

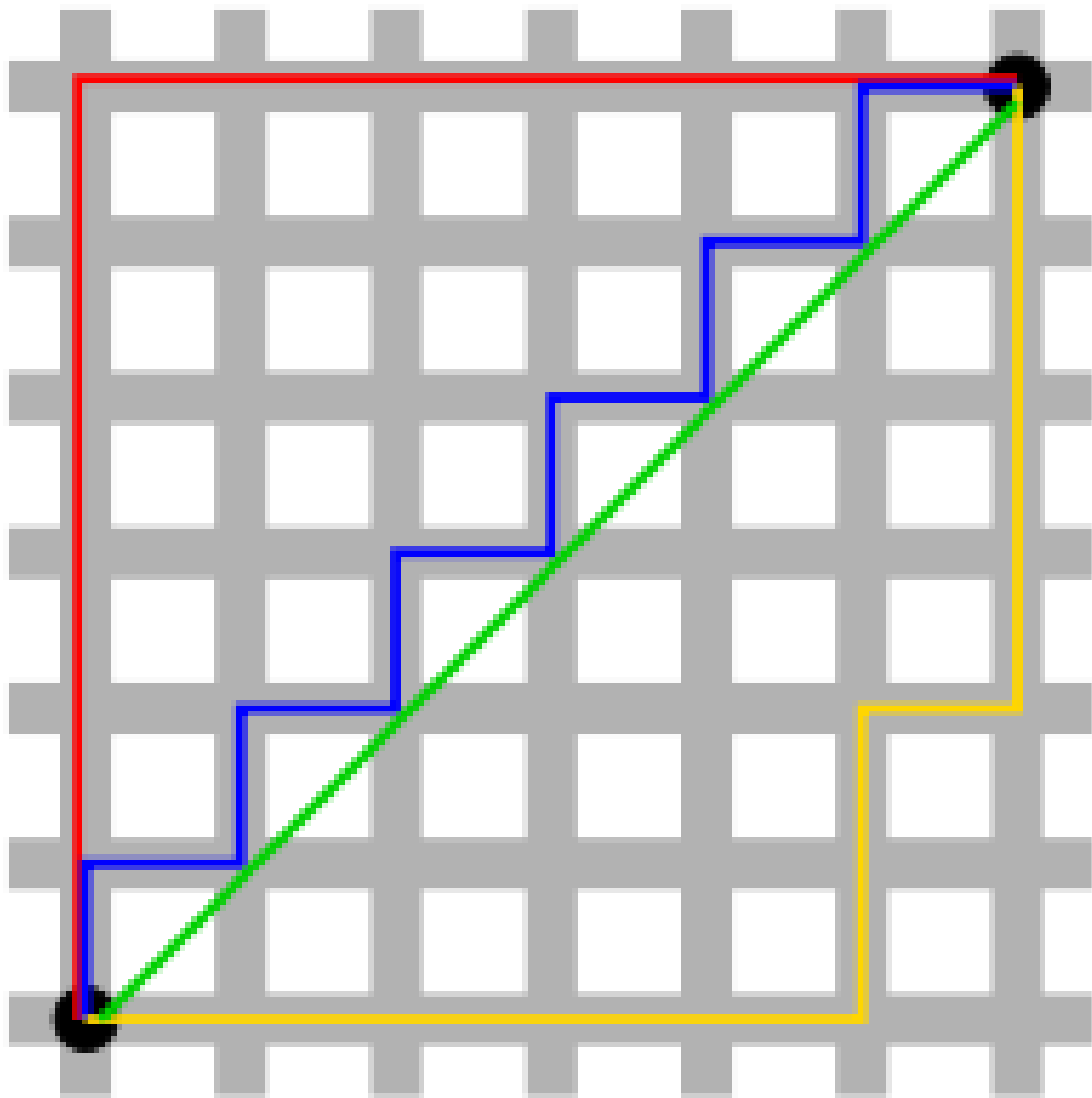
% empty_square_row(+Puzzle, ?Index) - Індекс (1) рядка з порожнім квадратом
empty_square_row(Puzzle, Index) :-
    nth0(I, Puzzle, 0),
    Index is I div 4 + 1,
    !.

pred(X, Y) :-
    Y < X,
    Y > 0.

% sum_row_inversions(+Board, -Count) - Count - це кількість інверсій для певної дошки
sum_row_inversions([], 0).
sum_row_inversions([X|Xs], R) :-
    include(pred(X), Xs, Ys),
    length(Ys, InversionCount),
    sum_row_inversions(Xs, Sum),
    R is InversionCount + Sum.
```

Знаходимо відстань інверсій

```
sum_row_inversions([X|Xs], R) :-  
    include(pred(X), Xs, Ys),  
    length(Ys, InversionCount),  
    sum_row_inversions(Xs, Sum),  
    R is InversionCount + Sum.  
  
% sum_column_inversions(+Board, -Count) - Count - це кількість інверсій для певної дошки  
sum_column_inversions([], 0).  
sum_column_inversions([X|Xs], R) :-  
    include(pred2(X), Xs, Ys),  
    length(Ys, InversionCount),  
    sum_column_inversions(Xs, Sum),  
    R is InversionCount + Sum.  
  
% inversion_distance(+Board, -N) - N є результатом евристичної інверсійної відстані для Board  
inversion_distance(State, Result) :-  
    sum_row_inversions(State, V),  
    divmod(V, 3, X, Y),  
    Vertical is X + Y,  
    transpose(State, StateT),  
    sum_column_inversions(StateT, H),  
    divmod(H, 3, X2, Y2),  
    Horizontal is X2 + Y2,  
    Result is Vertical + Horizontal.
```



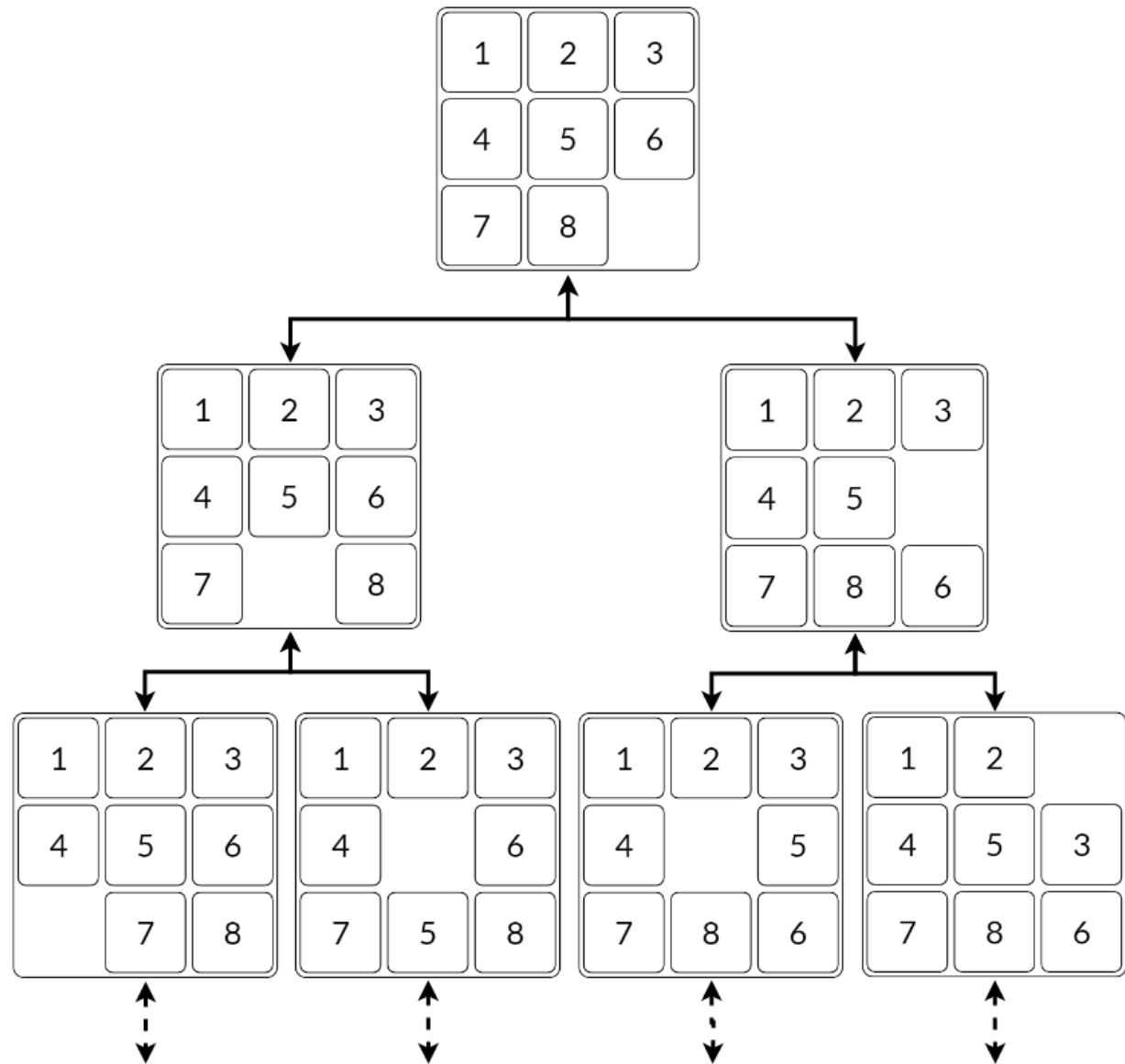
Манхетенська
відстань

Манхетенська відстань

```
% manhattan(+Board, -N) - N є результатом манхетенської відстані евристичної ф-ції для Board
manhattan(_, _, 0, 0).
manhattan(Current, Final, Num, Sum) :-
    nth0(Index, Current, Num),
    nth0(FinalIndex, Final, Num),
    index(Index, X1, Y1),
    index(FinalIndex, X2, Y2),
    Num1 is Num - 1,
    manhattan(Current, Final, Num1, Sum1),
    Sum is abs(X1 - X2) + abs(Y1 - Y2) + Sum1.

heuristic(Current, _, Res) :-
    manhattan(Current, M),
    inversion_distance(Current, I),
    Res is max(M, I).
```

Пошуковий алгоритм IDA*



Використання

```
solve_astar(Current, Final, Limit, Moves) :-  
    solve_astar(Current, Final, [Current], [], Limit, Moves).  
  
solve_astar(Final, Final, _, MovesBackwards, Limit, Moves) :-  
    Limit >= 0,  
    reverse(MovesBackwards, Moves).  
  
solve_astar(Current, Final, StateAcc, MovesBackwards, Limit, Moves) :-  
    heuristic(Current, Final, H),  
    Limit >= H,  
    L1 is Limit - 1,  
    move(Current, NewState, Direction),  
    \+member(NewState, StateAcc),  
    solve_astar(NewState, Final, [NewState|StateAcc], [Direction|MovesBackwards], L1, Moves).  
  
% solve_idastar(+Current, +Final, -Moves) - Пошук IDA* від Current до Final  
solve_idastar(Current, Final, Moves):-  
    heuristic(Current, Final, H),  
    between(H, 80, Limit),  
    solve_astar(Current, Final, Limit, Moves).
```


Розв'язок

```
solve(Board, Moves) :-  
    solvable(Board),  
    solve_idastar(Board, [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,0], Moves).
```



Підключення графічного інтерфейсу



```
:- http_handler('/solvable', handle_solvable, [methods([post, options])]).
:- http_handler('/solve', handle_solve, [methods([post, options])]).

start_server(Port) :-
    http_server(http_dispatch, [port(Port)]).

handle_solvable(Request) :-
    http_read_json_dict(Request, Dict),
    Puzzle = Dict.get(puzzle),
    (solvable(Puzzle) -> Response = json{result: true} ; Response = json{result: false}),
    cors_enable(Request, [methods([post, options])]),
    reply_json(Response).

handle_solve(Request) :-
    http_read_json_dict(Request, Dict),
    Board = Dict.get(board),
    (solve(Board, Moves) -> Response = json{result: Moves} ; Response = json{error: "Unsolvable"}),
    reply_json(Response).
```

Підключення графічного інтерфейсу

```
const express : e | () => core.Express = require('express');

const app : any | Express = express();
const port : number = 3000;

app.get('/', (req : Request<P, ResBody, ReqBody, ReqQuery, LocalsObj> , res : Response<ResBody, LocalsObj> ) : void => {
    res.sendFile( path: __dirname + '/start.html');
});

app.get('/game', (req : Request<P, ResBody, ReqBody, ReqQuery, LocalsObj> , res : Response<ResBody, LocalsObj> ) : void => {
    res.sendFile( path: __dirname + '/game.html');
});

app.listen(port, hostname: () : void => {
    console.log(`Server is running on port ${port}`);
});
```

Запускаємо два сервери

Сервер прологу на 8080

```
cd path/to/file
```

```
swipl
```

```
consult('server.pl').
```

Сервер node на 3000

```
cd path/to/file
```

```
node server.js
```

Заходимо в браузер на порт localhost:3000/

N-Puzzle Game

Choose Puzzle Size:

Start Game

N-Puzzle Game

Choose Puzzle Size:

4x4

3x3

4x4

5x5

N-Puzzle Game

Enter Values:

1 2 3 4 5...

Apply

1		7
8	6	5
2	3	4

Shuffle

Solve

Solvable?

Timer: 2 seconds

Steps: 0

N-Puzzle Game

Enter Values:

1 2 3 4 5...

Apply

3	23		8	2
18	17	6	1	16
7	24	20	19	5
14	9	4	10	22
11	12	21	13	15

Shuffle

Solve

Solvable?

Timer: 1 seconds

Steps: 0

N-Puzzle Game

Enter Values:

1 2 3 4 5...

Apply

5	12	8	
15	1	2	10
4	3	13	14
11	9	7	6

Shuffle

Solve

Solvable?

Timer: 1 seconds

Steps: 0

Повідомлення з localhost:3000

This puzzle is not solvable. Please shuffle the puzzle and try again.

OK

Повідомлення з localhost:3000

This puzzle is solvable.

OK



Shuffle

Solve

Solvable?

Timer: 2 seconds

До натиснення Apply вводимо значення

N-Puzzle Game

Enter Values:

1 10 15 4 13 6 3 8 2 9 12 7 14 5 0 11

Apply

8	14	12	1
11	4	13	9
2	3	5	10
	7	6	15

Shuffle

Solve

Solvable?

Timer: 00 seconds

Після

N-Puzzle Game

Enter Values:

1 10 15 4 13 6 3 8 2 9 12 7 14 5 0 11

Apply

1	10	15	4
13	6	3	8
2	9	12	7
14	5		11

Shuffle

Solve

Solvable?

Timer: 4 seconds

```

async function solve() {

  await fetch('http://localhost:3000/solve', {
    method: 'POST',
    headers: {
      'Content-Type': 'application/json',
    },
    body: JSON.stringify({ board: puzzleState }),
  })
  .then(response => {
    if (!response.ok) {
      throw new Error(`HTTP error! Status: ${response.status}`);
    }
    return response.json();
  })
  .then(data => {
    console.log('Server Response:', data);

    if (data && data.result) {
      animateMoves(data.result);
    } else {
      console.error('Error: Unexpected server response format. Missing "result"
property.');
```

alert('Error: Unexpected server response format. Please check the console for
details.');

```

    }
  })
  .catch(error => {
    console.error('Error solving puzzle:', error);

    alert('Error solving puzzle. Please check the console for details.');
```

});

```

  }
}

```

Solve

Congratulations!



You solved the puzzle in 79 seconds with 117 steps!

Again

Висновки

- що було цікавим?
- що вразило?
- що нового дізнались та чому новому навчились?
- з якими труднощами/складнощами зіткнулися? як їх подолали?
- яка трудомісткість роботи над проєктом та його складовими?
- чим була корисна робота над проєктом?