

KELOMPOK X

8 PUZZLE & 8 QUEENS

IMPLEMENTASI BFS DAN DFS

KECERDASAN BUATAN F



KELOMPOK X

Line 1
Line 2
Line 3

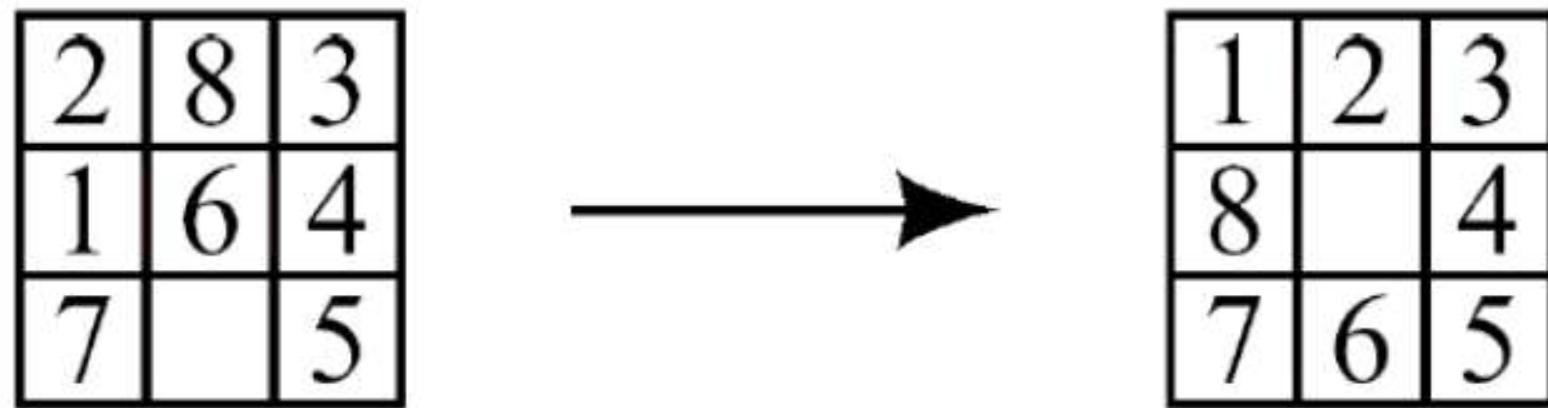
NADIF MUSTAFA / 5025211127
FREDERICK YONATAN / 5025211121
NIZAM HAKIM / 5025211209

KECERDASAN BUATAN F



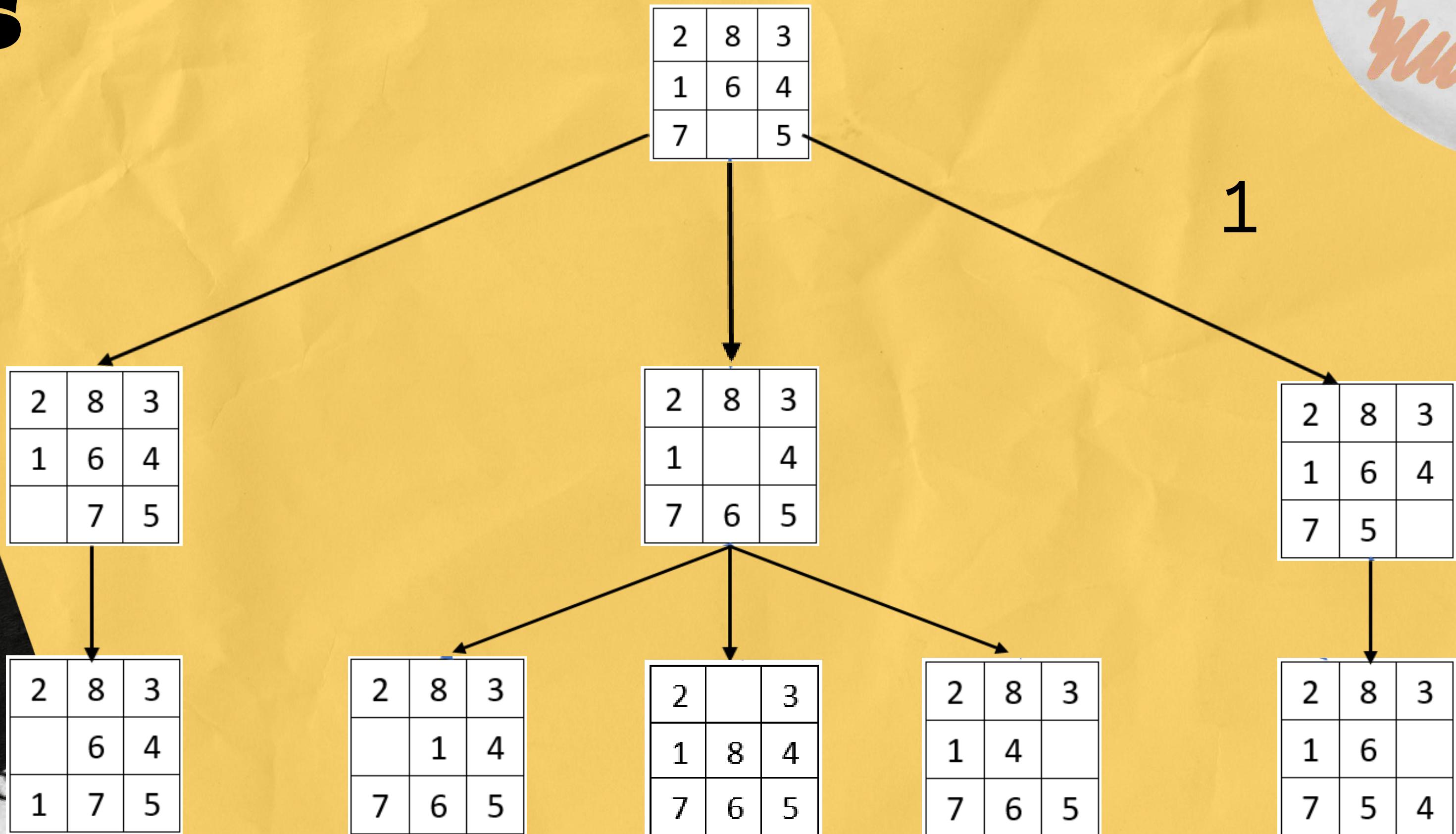
8 PUZZLE PROBLEM

- Terdapat kasus 8-puzzles dengan informasi state awal dan goal sebagai berikut:



- Selesaikan kasus 8-puzzles diatas menggunakan 3 metode *uninformed search*

BFS



2	8	3
6	4	
1	7	5

2	8	3
1	4	
7	6	5

2		3
1	8	4
7	6	5

2	8	3
1	4	
7	6	5

2	8	3
1	6	
7	5	4

3

	8	3
2	6	4
1	7	5

2	8	3
6		4
1	7	5

	8	3
2	1	4
7	6	5

2	8	3
7	1	4
	6	5

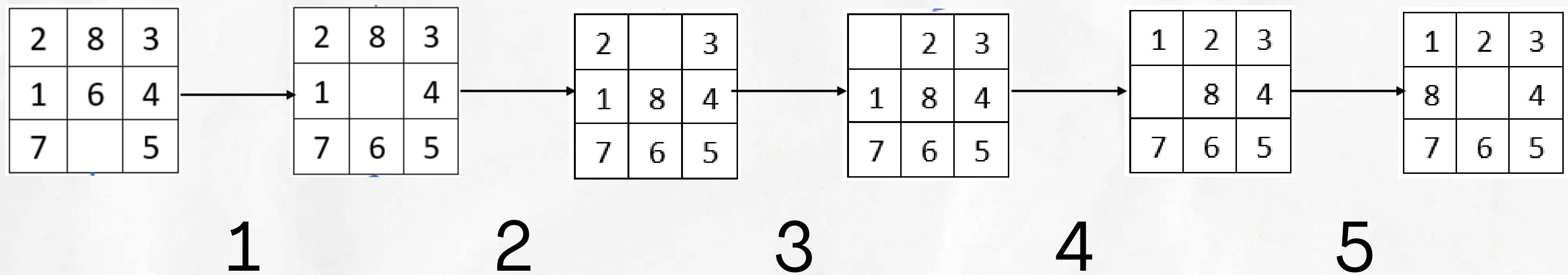
	2	3
1	8	4
7	6	5

2	3	
1	8	4
7	6	5

2	8	
1	4	3
7	6	5

2	8	3
1	4	5
7	6	

HASIL AKHIR BFS



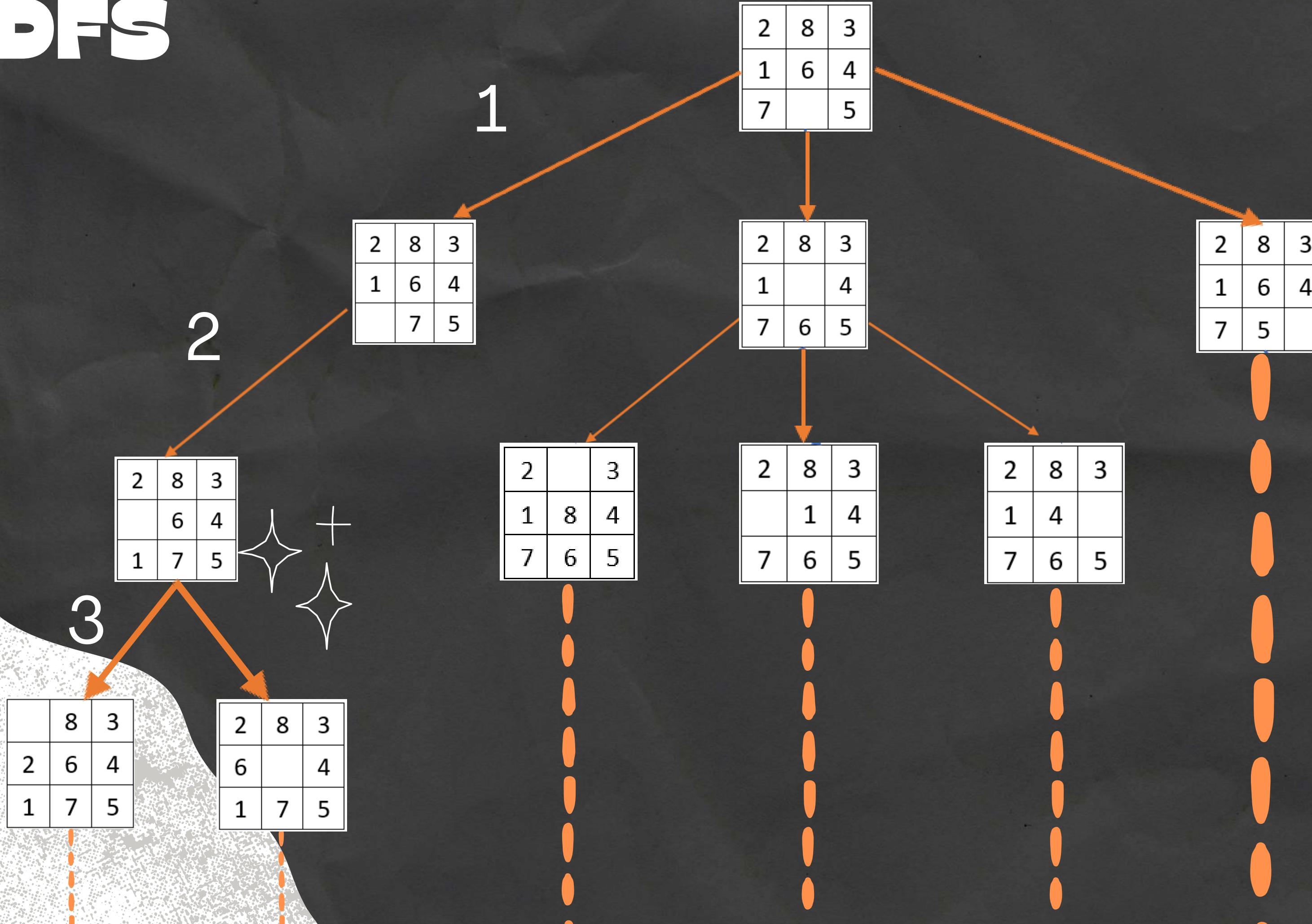
```
1 #include <bits/stdc++.h>
2 using namespace std;
3 #define mp make_pair
4 #define fr front
5 #define fi first
6 #define sc second
7 #define endl '\n'
8
9 set<string> vis;
10
11 int getPos(int idx, int dest)
12 {
13     int row = idx / 3;
14     int col = idx % 3;
15
16     if(dest == 0)
17         return (col == 0 ? -1 : 3 * row + col - 1); // Left
18     else if(dest == 1)
19         return (col == 2 ? -1 : 3 * row + col + 1); // Right
20     else if(dest == 2)
21         return (row == 0 ? -1 : 3 * (row - 1) + col); // Top
22     else
23         return (row == 2 ? -1 : 3 * (row + 1) + col); // Bottom
24 }
```

```
26 void bfs(string start, string finish)
27 {
28     queue<pair<string, int>> q;
29     q.push(mp(start, 0));
30
31     while(!q.empty())
32     {
33         string cur = q.front().fi;
34         int step = q.front().sc;
35
36         for(int i = 0; i <= 2; ++i){
37             for(int j = 0; j <= 2; ++j){
38                 cout << cur[3*i+j] << " ";
39             }
40             cout << endl;
41         }
42         cout << "step : " << step << endl << endl;
43
44         if(!vis.count(cur))
45             vis.insert(cur);
46         q.pop();
47     }
48 }
```

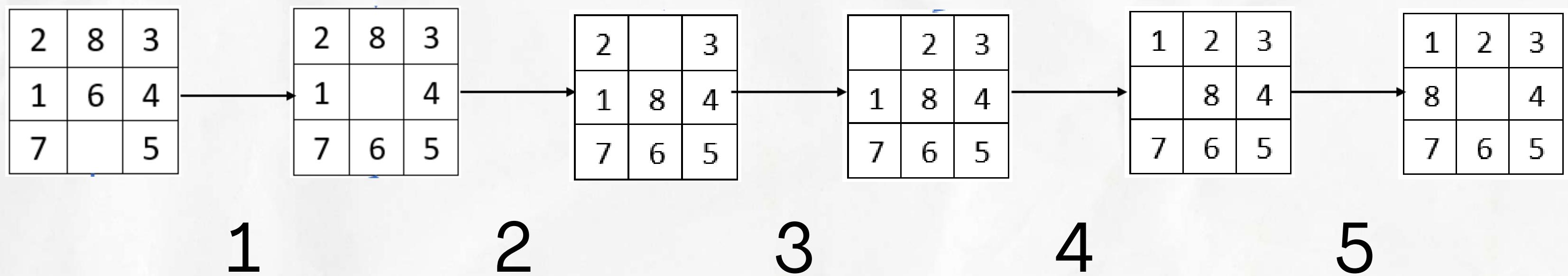
```
47
48     if(cur == finish){
49         cout << step << endl;
50         return;
51     }
52     int kosong = -1;
53
54     for(int i = 0; i < 9; ++i){
55         if(cur[i] == '0'){
56             kosong = i;
57             break;
58         }
59     }
60     for(int i = 0; i < 4; ++i)
61     {
62         string nextStr = cur;
63         int nextPos = getPos(kosong, i);
64
65         if(nextPos != -1){
66             swap(nextStr[kosong], nextStr[nextPos]);
67         }
68         if(!vis.count(nextStr))
69         {
70             vis.insert(nextStr);
71             q.push(mp(nextStr, step + 1));
72         }
73     }
```

```
77 int main()
78 {
79     ios_base::sync_with_stdio(false);
80     cin.tie(NULL);
81     cout.tie(NULL);
82
83     string start = "283164705";
84     string finish = "123804765";
85
86     bfs(start, finish);
87     return 0;
88 }
```

DFS



HASIL AKHIR DFS



```
1 #include <bits/stdc++.h>
2 using namespace std;
3 #define mp make_pair
4 #define fi first
5 #define sc second
6 #define endl '\n'
7
8 set<string> vis;
9
10 int getPos(int idx, int dest)
11 {
12     int row = idx / 3;
13     int col = idx % 3;
14
15     if(dest == 0)
16         return (col == 0 ? -1 : 3 * row + col - 1);      // Left
17     else if(dest == 1)
18         return (col == 2 ? -1 : 3 * row + col + 1);      // Right
19     else if(dest == 2)
20         return (row == 0 ? -1 : 3 * (row - 1) + col);    // Top
21     else
22         return (row == 2 ? -1 : 3 * (row + 1) + col);    // Bottom
23 }
```

```
25 void dfs(string start, string finish)
26 {
27     stack<pair<string, int>> st;
28     st.push(mp(start, 0));
29
30     while(!st.empty())
31     {
32         string cur = st.top().fi;
33         int step = st.top().sc;
34
35         for(int i = 0; i <= 2; ++i){
36             for(int j = 0; j <= 2; ++j){
37                 cout << cur[3*i+j] << " ";
38             }
39             cout << endl;
40         }
41         cout << "step : " << step << endl << endl;
42
43         if(!vis.count(cur))
44             vis.insert(cur);
45         st.pop();
46     }
47 }
```

```
47    if(cur == finish){
48        cout << step << endl;
49        return;
50    }
51    int kosong = -1;
52
53    for(int i = 0; i < 9; ++i){
54        if(cur[i] == '0'){
55            kosong = i;
56            break;
57        }
58    }
59    for(int i = 0; i < 4; ++i)
60    {
61        string nextStr = cur;
62        int nextPos = getPos(kosong, i);
63
64        if(nextPos != -1){
65            swap(nextStr[kosong], nextStr[nextPos]);
66        }
67        if(!vis.count(nextStr))
68        {
69            vis.insert(nextStr);
70            st.push(mp(nextStr, step + 1));
71        }
72    }
73}
74}
```

```
76 int main()
77 {
78     ios_base::sync_with_stdio(false);
79     cin.tie(NULL);
80     cout.tie(NULL);|
81
82     string start = "283164705";
83     string finish = "123804765";
84
85     dfs(start, finish);
86     return 0;
87 }
```

BFS

```
2 4 8  
1 0 3  
7 6 5  
step : 5
```

```
2 8 3  
1 4 5  
0 7 6  
step : 5
```

```
2 8 3  
1 0 5  
7 4 6  
step : 5
```

```
1 2 3  
8 0 4  
7 6 5  
step : 5
```

DFS

```
1 3 4  
8 2 0  
7 6 5  
step : 28010
```

```
1 3 0  
8 2 4  
7 6 5  
step : 28011
```

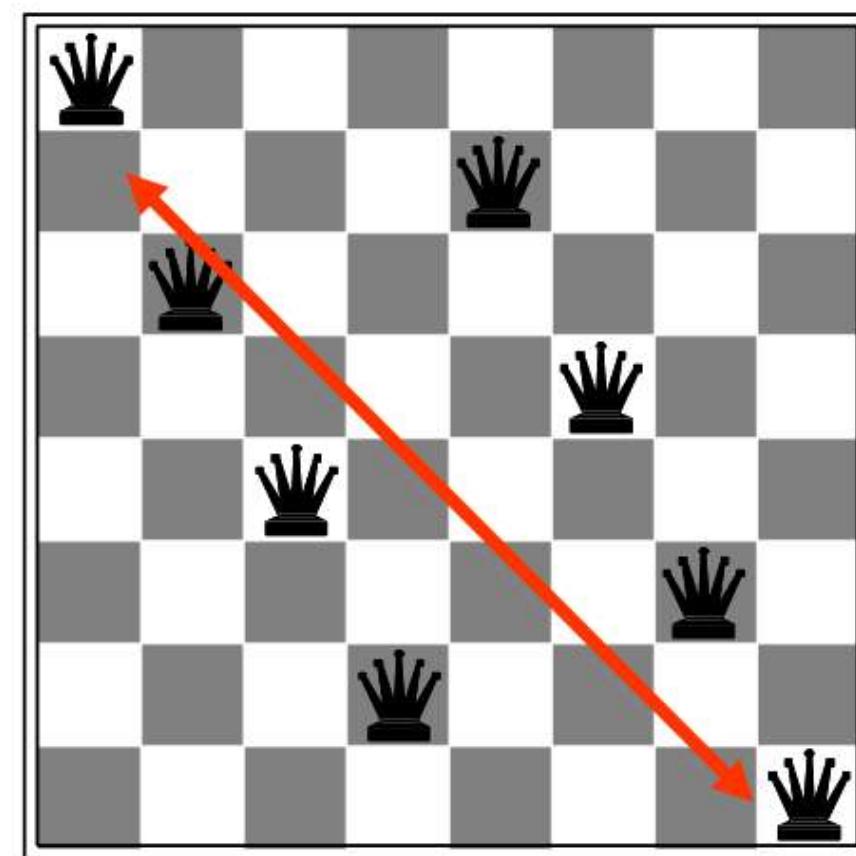
```
1 0 3  
8 2 4  
7 6 5  
step : 28012
```

```
1 2 3  
8 0 4  
7 6 5  
step : 28013
```

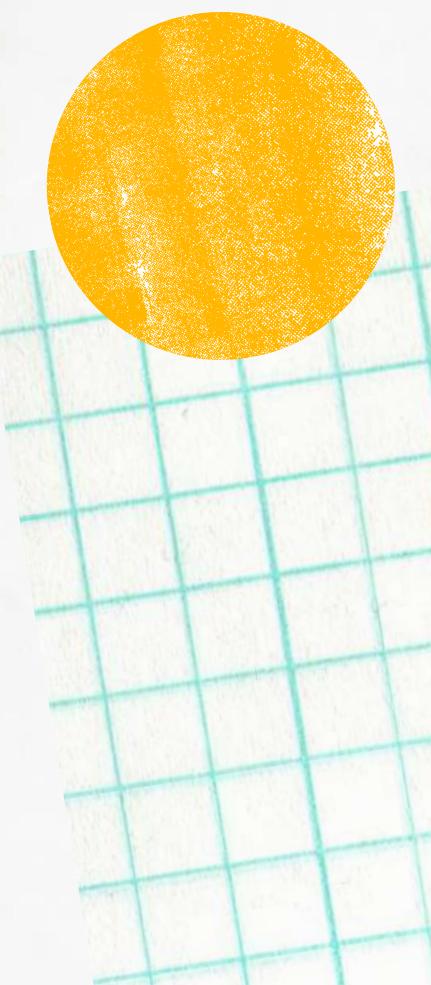
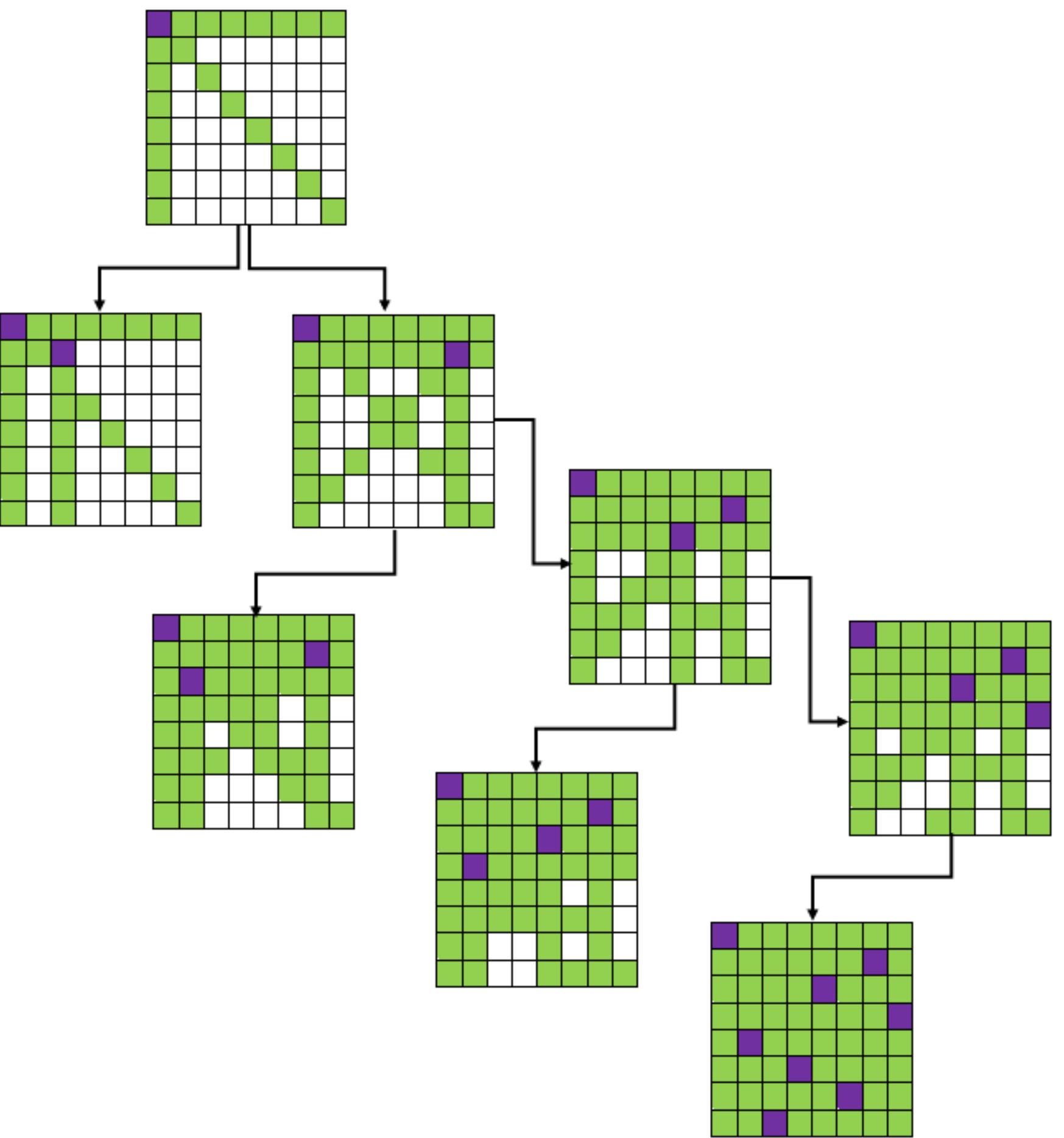
BFS VS DFS

8 QUEENS PROBLEM

- *State* : susunan 0..8 ratu pada papan catur
- *Initial State* : Tidak ada ratu pada papan catur
- *Successor Function* : Masukkan ratu ke papan catur
- *Goal Test* : Tidak ada ratu yang saling serang



ILUSTRASI BFS



BFS

```
1 #include <bits/stdc++.h>
2 using namespace std;
3 #define fr front
4 #define tab '\t'
5 #define endl '\n'
6
7 typedef struct
8 {
9     string str;
10    int col, numQueens;
11 }
12 state;
13
14 state defState;
15 string defStr = "";
16 set<string> vis;
17 int solutions;
18
19 void init()
20 {
21     for(int i = 0; i < 64; ++i)
22         defStr += (char) 100;
23     defState.str = defStr;
24     defState.col = 0;
25     defState.numQueens = 0;
26 }
```

struct state untuk mendefinisikan state papan catur

fungsi init untuk menginisiasi kondisi awal papan catur (kosong)

```
28 ✓ void str_to_board(string str, int board[8][8])
29   {
30     int itr = 0;
31   ✓ for(int i = 0; i < 8; ++i){
32     ✓ for(int j = 0; j < 8; ++j)
33     {
34       board[i][j] = (int) (str[itr] - 100);
35       itr++;
36     }
37   }
38 }
39
40 ✓ string board_to_str(int board[8][8])
41 {
42   string res = "";
43   ✓ for(int i = 0; i < 8; ++i){
44     ✓ for(int j = 0; j < 8; ++j)
45       res += (char) (board[i][j] + 100);
46   }
47   return res;
48 }
```

str_to_board
mentranslate kondisi papan dari
string ke dalam array 2 dimensi board

board_to_str
mentranslate kondisi papan dari array
2 dimensi board ke string

fungsi marking menandai petak-petak yang terserang oleh queen

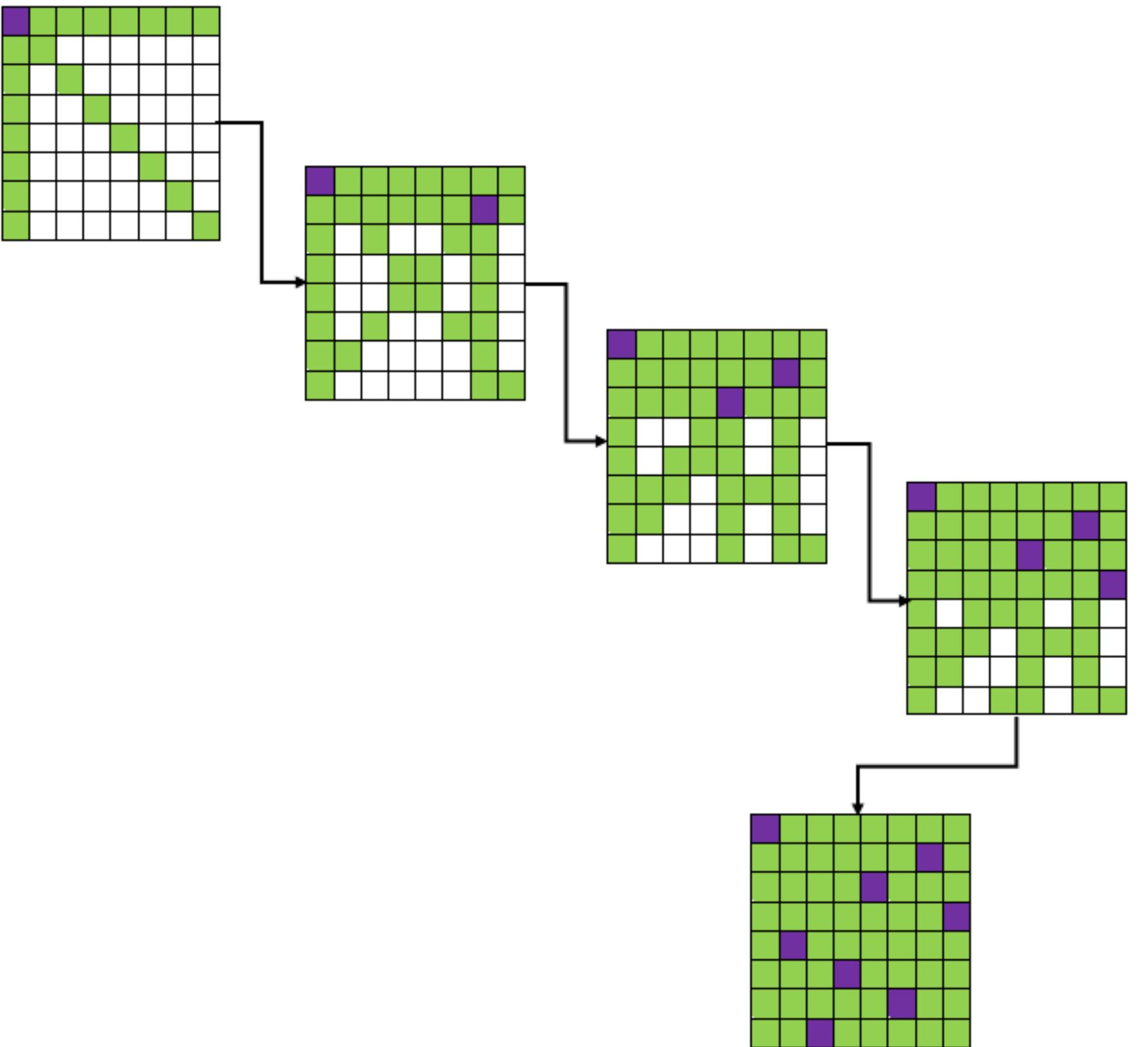
```
50 void marking(int board[8][8], int i, int col, int mark)
51 {
52     for(int j = 0; j < 8; ++j){
53         if(j != col) board[i][j] += mark;
54         if(j != i) board[j][col] += mark;
55     }
56     for(int j = 1; j < 8; ++j){
57         if(i - j >= 0 and col - j >= 0) board[i - j][col - j] += mark;
58         if(i + j < 8 and col + j < 8) board[i + j][col + j] += mark;
59         if(i - j >= 0 and col + j < 8) board[i - j][col + j] += mark;
60         if(i + j < 8 and col - j >= 0) board[i + j][col - j] += mark;
61     }
62 }
```

```
64 void bfs()
65 {
66     queue<state> q;
67     q.push(defState);
68
69     while(!q.empty())
70     {
71         state curState = q.fr();
72
73         int board[8][8];
74         str_to_board(curState.str, board);
75
76         if(!vis.count(curState.str))
77             vis.insert(curState.str);
78         q.pop();
79
80         if(curState.numQueens >= 8)
81         {
82             for(int i = 0; i < 8; ++i){
83                 for(int j = 0; j < 8; ++j)
84                     cout << board[i][j] << tab;
85                 cout << endl;
86             }
87             cout << endl;
88             solutions++;
89             continue;
90         }
91     }
92 }
```

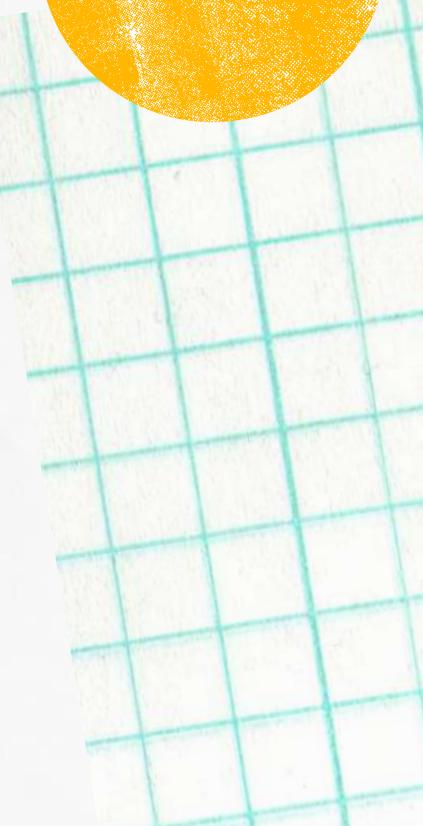
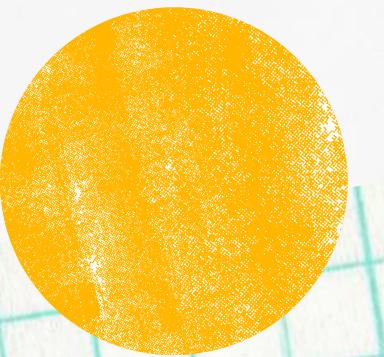
```
91     for(int i = 0; i < 8; ++i)
92     {
93         if(board[i][curState.col] > -1)
94         {
95             board[i][curState.col] = 1;
96
97             marking(board, i, curState.col, -1);
98             string nextStr = board_to_str(board);
99
100            if(!vis.count(nextStr))
101            {
102                vis.insert(nextStr);
103
104                state nextState = curState;
105                nextState.str = nextStr;
106                nextState.col++;
107                nextState.numQueens++;
108
109                q.push(nextState);
110            }
111        }
112    }
113 }
114 }
```

fungsi main

```
117 int main()
118 {
119     init();
120     bfs();
121     cout << solutions << endl;
122     return 0;
123 }
```



ILUSTRASI DFS



DFS

struct state untuk mendefinisikan state papan catur

fungsi init untuk menginisiasi kondisi awal papan catur (kosong)

```
1 #include <bits/stdc++.h>
2 using namespace std;
3 #define tab '\t'
4 #define endl '\n'
5
6 typedef struct
7 {
8     string str;
9     int col, numQueens;
10 }
11 state;
12
13 state defState;
14 string defStr = "";
15 set<string> vis;
16 int solutions;
17
18 void init()
19 {
20     for(int i = 0; i < 64; ++i)
21         defStr += (char) 100;
22     defState.str = defStr;
23     defState.col = 0;
24     defState.numQueens = 0;
25 }
26
```

str_to_board
mentranslate kondisi papan dari
string ke dalam array 2 dimensi board

board_to_str
mentranslate kondisi papan dari array
2 dimensi board ke string

```
27 void str_to_board(string str, int board[8][8])
28 {
29     int itr = 0;
30     for(int i = 0; i < 8; ++i){
31         for(int j = 0; j < 8; ++j)
32         {
33             board[i][j] = (int) (str[itr] - 100);
34             itr++;
35         }
36     }
37 }
38
39 string board_to_str(int board[8][8])
40 {
41     string res = "";
42     for(int i = 0; i < 8; ++i){
43         for(int j = 0; j < 8; ++j)
44             res += (char) (board[i][j] + 100);
45     }
46     return res;
47 }
```

```
49 void marking(int board[8][8], int i, int col, int mark)
50 {
51     for(int j = 0; j < 8; ++j){
52         if(j != col) board[i][j] += mark;
53         if(j != i) board[j][col] += mark;
54     }
55     for(int j = 1; j < 8; ++j){
56         if(i - j >= 0 and col - j >= 0) board[i - j][col - j] += mark;
57         if(i + j < 8 and col + j < 8) board[i + j][col + j] += mark;
58         if(i - j >= 0 and col + j < 8) board[i - j][col + j] += mark;
59         if(i + j < 8 and col - j >= 0) board[i + j][col - j] += mark;
60     }
61 }
```

fungsi marking menandai petak-petak yang terserang oleh queen

```
63 void dfs()
64 {
65     stack<state> st;
66     st.push(defState);
67
68     while(!st.empty())
69     {
70         state curState = st.top();
71
72         int board[8][8];
73         str_to_board(curState.str, board);
74
75         if(!vis.count(curState.str))
76             vis.insert(curState.str);
77         st.pop();
78
79         if(curState.numQueens >= 8)
80         {
81             for(int i = 0; i < 8; ++i){
82                 for(int j = 0; j < 8; ++j)
83                     cout << board[i][j] << tab;
84                     cout << endl;
85             }
86             cout << endl;
87             solutions++;
88             continue;
89         }

```

```
90         for(int i = 0; i < 8; ++i)
91         {
92             if(board[i][curState.col] > -1)
93             {
94                 board[i][curState.col] = 1;
95
96                 marking(board, i, curState.col, -1);
97                 string nextStr = board_to_str(board);
98
99                 if(!vis.count(nextStr))
100                 {
101                     vis.insert(nextStr);
102
103                     state nextState = curState;
104                     nextState.str = nextStr;
105                     nextState.col++;
106                     nextState.numQueens++;
107
108                     st.push(nextState);
109                 }
110             }
111         }
112     }
113 }
114 }
```

```
116 int main()
117 {
118     init();
119     dfs();
120     cout << solutions << endl;
121     return 0;
122 }
```

fungsi main

BFS VS DFS

BFS

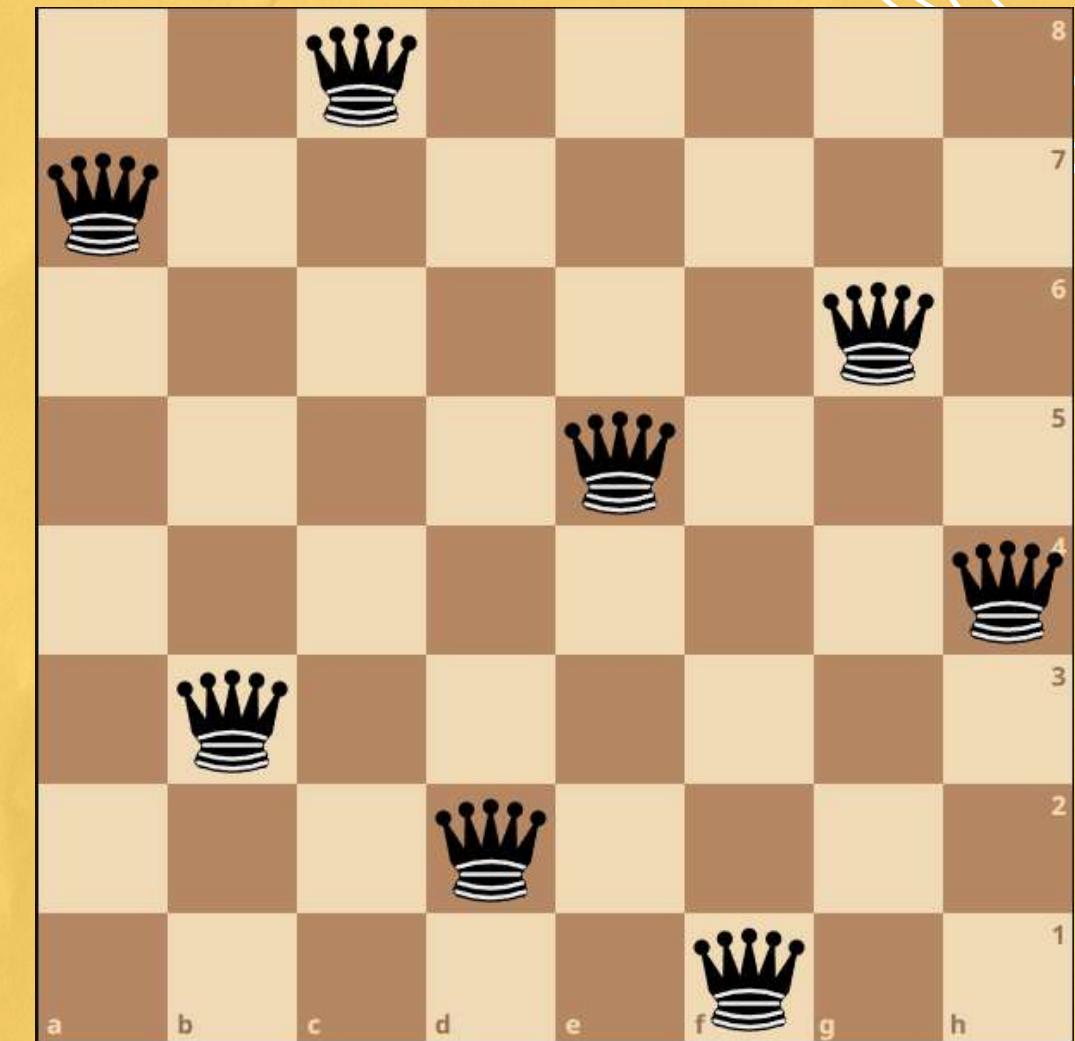
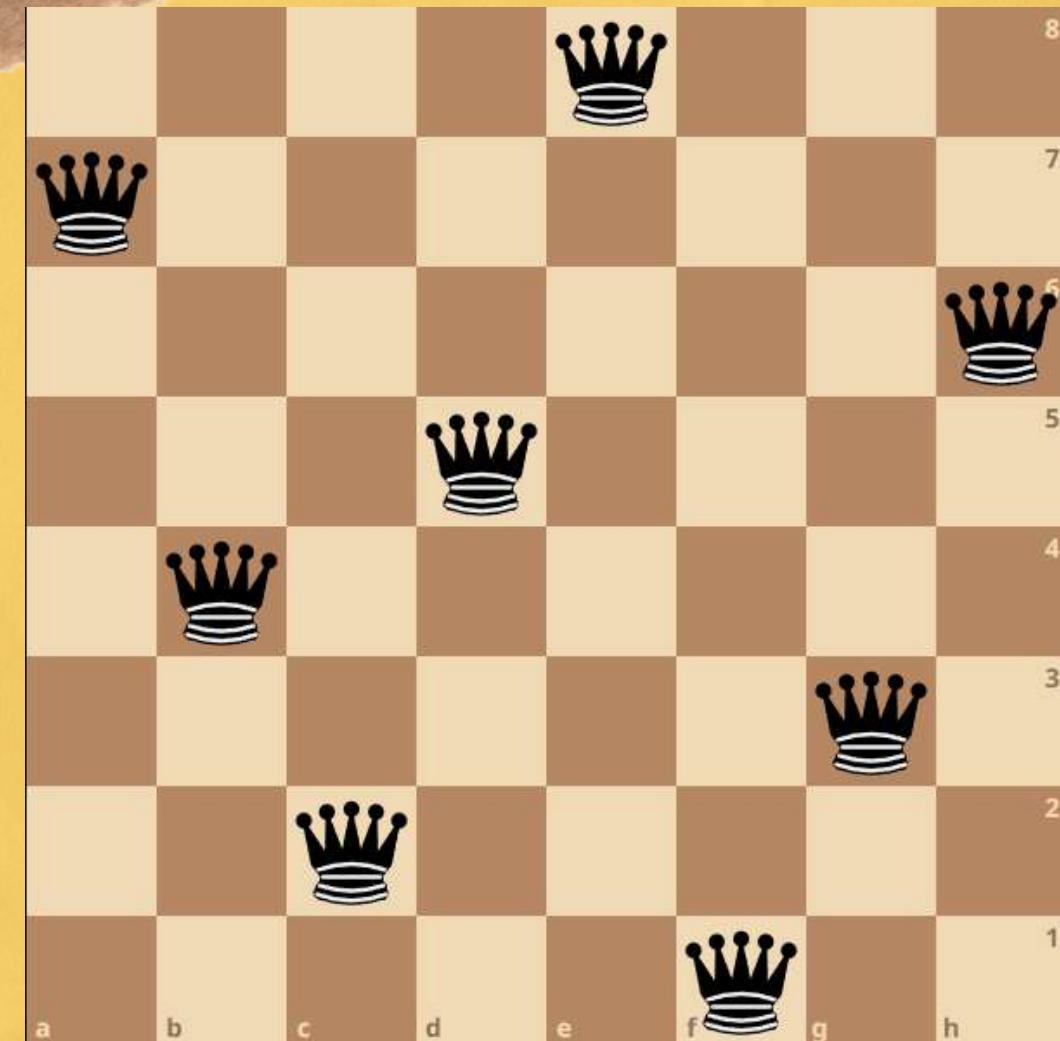
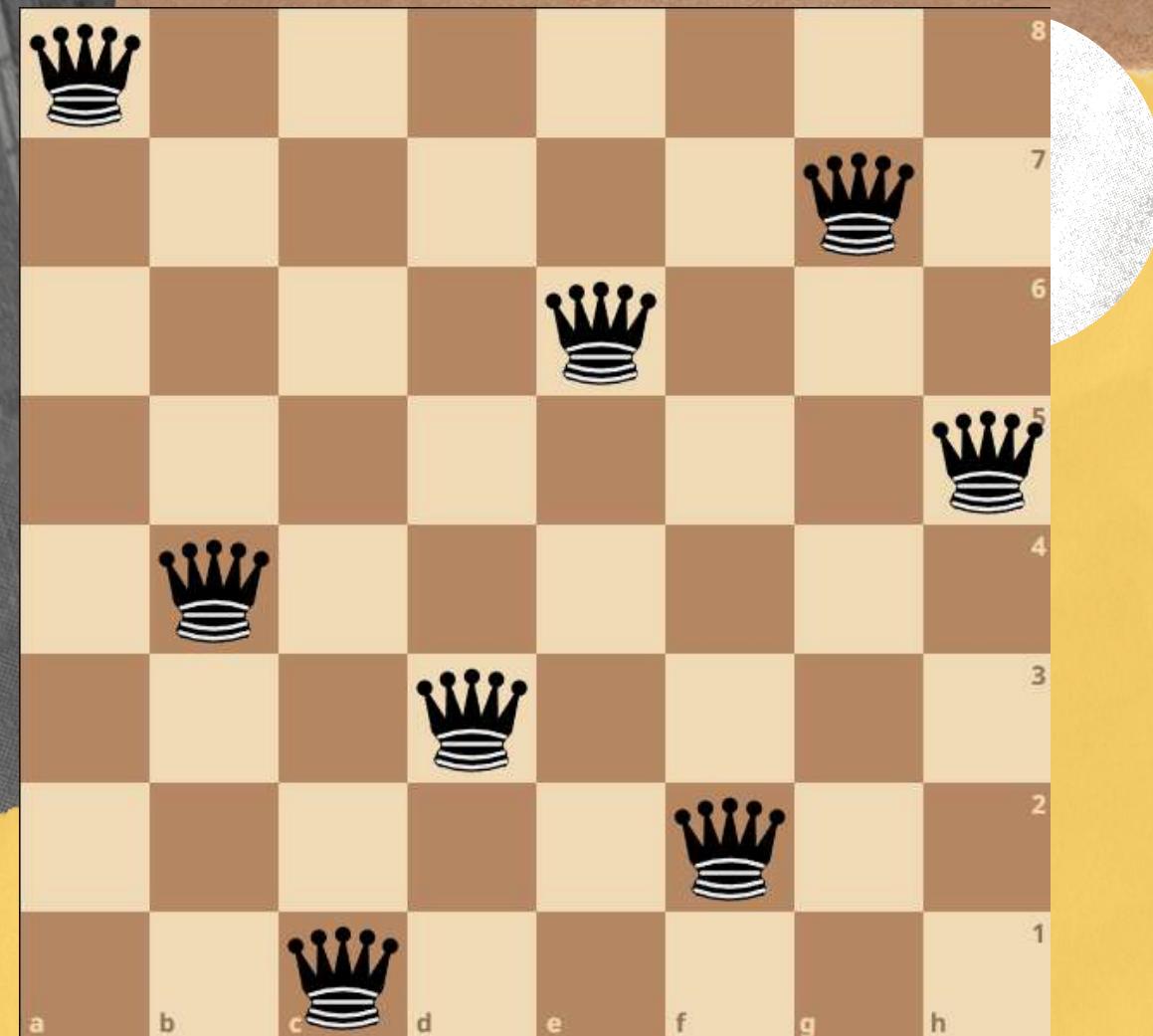
-2	-2	-4	-2	1	-3	-3	-3
-2	1	-3	-4	-4	-3	-2	
-3	-3	-4	1	-4	-4	-3	-3
-1	-4	-3	-4	-4	-2	1	-3
-2	-3	1	-4	-3	-4	-3	-4
-2	-3	-3	-3	-4	-3	-4	1
-2	-3	-2	-3	-3	-4	-3	-2
1	-2	-3	-2	-3	-4	-3	-3
-2	-2	1	-4	-3	-3	-2	-3
-2	-3	-4	-3	1	-3	-3	
-3	1	-3	-4	-3	-4	-3	
-2	-3	-4	-2	-4	-4	1	
-1	-3	-3	-4	-4	1	-4	-3
-2	-2	-3	1	-4	-3	-4	
-1	-3	-3	-3	-4	1	-3	
1	-3	-3	-3	-2	-4	-3	-3
-2	-2	1	-3	-4	-3	-3	-3
-2	-3	-3	-4	-4	1	-3	-2
-3	-2	-4	1	-4	-4	-3	-3
-1	1	-4	-4	-3	-4	-3	
-2	-3	-4	-4	-2	-4	-3	1
-2	-3	-3	-3	1	-3	-4	-4
-2	-3	-2	-3	-3	-4	1	-3
1	-2	-3	-2	-3	-4	-3	-3

92

DFS

1	-2	-3	-2	-3	-4	-3	-3
-3	-3	-1	-2	-3	1	-4	-3
-3	-2	-3	-3	-3	-3	-4	1
-3	-2	1	-4	-3	-4	-3	-4
-2	-3	-4	-4	-4	-3	1	-3
-4	-2	-4	1	-4	-4	-3	-3
-3	1	-3	-4	-4	-4	-3	-2
-3	-3	-4	-3	1	-3	-3	-3
-3	-3	-3	-4	-3	1	-3	-3
1	-2	-3	-2	-3	-4	-3	-3
1	-3	-3	-3	-2	-4	-3	-3
-2	-3	-3	-2	-2	-2	-4	1
-3	-2	-1	-2	-2	-4	-3	-3
-3	-1	-3	1	-4	-4	-3	-4
-2	-2	-2	-2	-4	-4	1	-4
-3	-2	-4	-4	-3	-4	-3	-3
-3	-2	-4	-4	1	-4	-4	-3
-3	-3	-4	-3	-2	-4	1	-3
-3	-3	-3	-4	-2	-4	1	-3
-3	-3	-3	-4	1	-3	-4	-3
-3	-3	-3	-4	-3	1	-3	-3
1	-2	-3	-2	-3	-4	-3	-3

92



CONTOH
SOLUSI



TERIMA KASIH

