2CSDE85 - Artificial Intelligence

Practical 3

Author: Darshil Maru 20BCE514

Instructor: Prof. Jigna Patel

Date: September 16, 2022

AIM: Program to implement hill climbing (for 8 queen problem)

Code:

```
#include <iostream>
#include <random>
#include <time.h>

using namespace std;

void printState(int st[])
{

    for (int i = 0; i < 4; i++)
    {
        for (int j = 0; j < 4; j++)
        {
            if (i == st[j])
            {
                cout << "1 ";
            }
            else
                cout << "0 ";
        }
        cout << endl;
    }
}

int conflict(int state[]) // count number of conflict in any given state
{
    int ans = 0;

    for (int i = 0; i < 4; i++)
    {
        int k = state[i];

        int count = 1;
```

```
        for (int j = i + 1; j < 4; j++, count++)
        {
            if (state[j] == k || state[j] == k - count || state[j] == k +
count)
                ans++;
        }

        // cout<<i<<" "<<ans<<endl;
    }
    return ans;
}

int main()
{
    srand(time(0));
    int i = 5;

    int board[4][4];
    int state[4];

    for (int i = 0; i < 4; i++)
    {
        state[i] = rand() % 4;
    }

    cout << "\nInitial State:" << endl;

    for (int i = 0; i < 4; i++)
        cout << state[i] << " ";
    cout << endl
         << endl;

    printState(state);
    cout << conflict(state) << " .. " << endl;

    int count = 0;

    while (conflict(state) != 0 && count < 100)
    {
        for (int i = 0; i < 4; i++) // for all states
```

```
        {
            bool move = false;
            for (int j = 0; j < 4; j++)
            {
                int state_move[4];
                for (int k = 0; k < 4; k++)
                    state_move[k] = state[k];
                state_move[i] = j;
                if (conflict(state) > conflict(state_move))
                {
                    state[i] = j;
                    move = true;
                    break;
                }
            }
            if (conflict(state) == 0)
                break;
            // plateue situation
            if (!move)
            {
                for (int j = 0; j < 4; j++)
                {
                    int state_move[4];
                    for (int k = 0; k < 4; k++)
                        state_move[k] = state[k];
                    state_move[i] = j;
                    if (conflict(state) == conflict(state_move) &&
state[i] != j)
                    {
                        state[i] = j;
                        move = true;
                        break;
                    }
                }
            }
        }
        if (count < 5)
        {
            printState(state);
            cout << endl;
```

```
        }
        count++;
    }

    printState(state);
    cout << endl;
    cout << conflict(state) << " " << count << endl;

    return 0;
}
```

Output:

```
PS C:\Users\Admin> cd "e:\Google Downloads\" ; if ($?) { g++ Nqueen.cpp -o Nqueen } ; if ($?) { .\Nqueen }

1 0 0 0
0 0 1 0
0 1 0 0

0 0 1 0
1 0 0 0
0 0 0 1
0 1 0 0

0 0 1 0
1 0 0 0
0 0 0 1
0 1 0 0

0 2
PS E:\Google Downloads>
```