



Problema celor 8 regine

TEMA DE PROIECT

Marcu Gabriela | Metode avansate de programare | 19.12.2022

Enunțul Problemei

Să se scrie un program care plasează 8 regine pe tabla de șah, fără ca acestea să se atace reciproc. Problema a fost investigată de Carl Friedrich Gauss în 1850 și până astăzi aceasta nu are o soluție analitică ci prin încercări. Problema are 92 de soluții din care 12 sunt distincte (din motive de simetrie).

Motivarea și scopul problemei

Fiind dată o tablă de șah, de dimensiune $n \times n$, se cer toate soluțiile de aranjare a n dame, astfel încât să nu se afle două dame pe aceeași linie, coloană sau diagonală (dame să nu se atace reciproc).

Pentru reprezentarea unei soluții putem folosi un vector cu n componente (având în vedere că pe fiecare linie se găsește o singură damă). De exemplu pentru soluția găsită avem vectorul ST ce poate fi asimilat unei stive. Două dame se găsesc pe aceeași diagonală dacă și numai dacă este îndeplinită condiția: $|st(i) - st(j)| = |i - j|$ (diferența, în modul, între linii și coloane este aceeași).

$ST(4)$

$ST(3)$ În general $ST(i) = k$ semnifică faptul că pe linia i dama ocupă poziția k .

$ST(2)$

$ST(1)$

Exemplu: în tabla 4×4 avem situația:

$$st(1) = 1 \quad i = 1$$

$$st(3) = 3 \quad j = 3$$

$$|st(1) - st(3)| = |1 - 3| = 2$$

$$|i - j| = |1 - 3| = 2$$

	D		
			D
D			
		D	

sau situația

$$st(1) = 3 \quad i = 1$$

$$st(3) = 1 \quad j = 3$$

$$|st(i) - st(j)| = |3 - 1| = 2$$

$$|i - j| = |1 - 3| = 2$$

	D		
			D
D			
		D	

Întrucât două dame nu se pot găsi în aceeași coloană, rezultă că o soluție este sub formă de permutare. O primă idee ne conduce la generarea tuturor permutărilor și la extragerea soluțiilor pentru problema ca două dame să nu fie plasate în aceeași diagonală. A proceda astfel, înseamnă că lucrăm conform strategiei *backtracking*. Aceasta presupune ca imediat ce am găsit două dame care se atacă, să reluăm căutarea.

lăță algoritmul, conform strategiei generate de *backtracking*:

- În prima poziție a stivei se încarcă valoarea 1, cu semnificația că în linia unu se așază prima damă în coloană.

- Linia 2 se încearcă așezarea damei în coloana 1, acest lucru nefiind posibil întrucât avem două dame pe aceeași coloană.

- În linia 2 se încearcă așezarea damei în coloana 2, însă acest lucru nu este posibil, pentru că damele se găsesc pe aceeași diagonală ($|st(1)-st(2)|=|1-2|$); - Așezarea damei 2 în coloana 3 este posibilă.

- Nu se poate plasa dama 3 în coloana 1, întrucât în liniile 1-3 damele ocupa aceeași coloană.

- Și această încercare eșuează întrucât damele de pe 2 și 3 sunt pe aceeași diagonală.

- Damele de pe 2-3 se găsesc pe aceeași coloană.

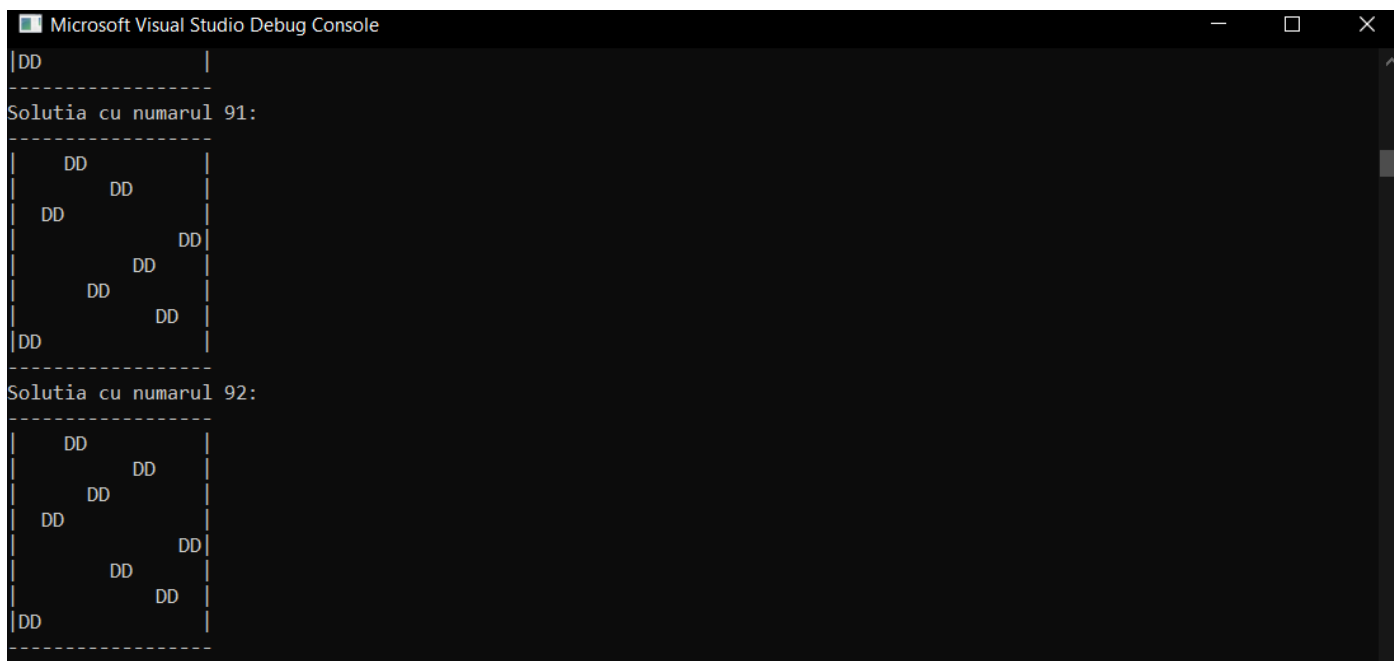
- Damele de pe 2-3 se găsesc pe aceeași diagonală. - Am coborât în stivă mutând dama de pe linia 2 și coloana 3 în coloana 4.

Algoritmul se încheie atunci când stiva este vidă. Semnificația procedurilor utilizate este următoarea:

VALID - validează valoarea pusă pe nivelul k al stivei, verificând dacă nu avem două dame pe aceeași linie ($st(k)=st(i)$), sau dacă nu avem două dame pe aceeași diagonală ($st(k)-st(i)=|k-i|$) caz în care returnează 1; în caz contrar, returnează 0; SOLUTIE - verifică dacă stiva a fost completată până la nivelul n inclusiv;

TIPAR - tipărește o soluție.

Rezultatul obținut :



```
Microsoft Visual Studio Debug Console
|DD|
-----
Solutia cu numarul 91:
|DD|
|DD|
|DD|
|DD|
|DD|
|DD|
|DD|
|DD|
-----
Solutia cu numarul 92:
|DD|
|DD|
|DD|
|DD|
|DD|
|DD|
|DD|
|DD|
-----
```

Soluția implementată în C++ :

```
#include<iostream>
#include<math.h>
using namespace std;

int n, st[25], nr_sol = 0 , a[25][25];

int valid(int k)
{
    for (int i = 1; i < k; i++)
    {
        if (st[i] == st[k])
            return 0;
        if (abs((float)(st[k] - st[i])) == abs((float)(k - i)))
            return 0;
    }
    return 1;
}

void tipar(int k)
{
    nr_sol++;
    cout << "Solutia cu numarul" << nr_sol << ": "; cout << endl;
    cout << "-";
    for (int i = 1; i <= n; i++)
        cout << "--";
    cout << "-\n";
    for (int i = 1; i <= n; i++)
    {
        cout << "|";
        for (int j = 1; j <= n; j++)
            if (st[j] == i) cout << "DD"; else cout << " ";
        cout << "|\n";
    }
    cout << "-";
    for (int i = 1; i <= n; i++)
        cout << "--";
    cout << "-\n";
}

void back()
{
    int k;
    k = 1;
    st[1] = 0;
    while (k > 0)
    {
        if (st[k] < n)
        {
            st[k]++; //urcam in stiva <=> trecem la urm coloana
            if (valid(k))
            {
                if (k == n)
                    tipar(k);
                else
                    k = k + 1;
            }
        }
        else
        {
            st[k] = 0;
            k--;
        }
    }
}

int main()
{
    do
    {
        cout << "Dati numarul de regine de pe tabla de sah: ";
        cin >> n;
    } while (n <= 3);
    back();
    return 0; }
```

Bibliografie

<http://www.panseluta.ro/c/regine.pdf>