Problema Reginei N
Backtracking

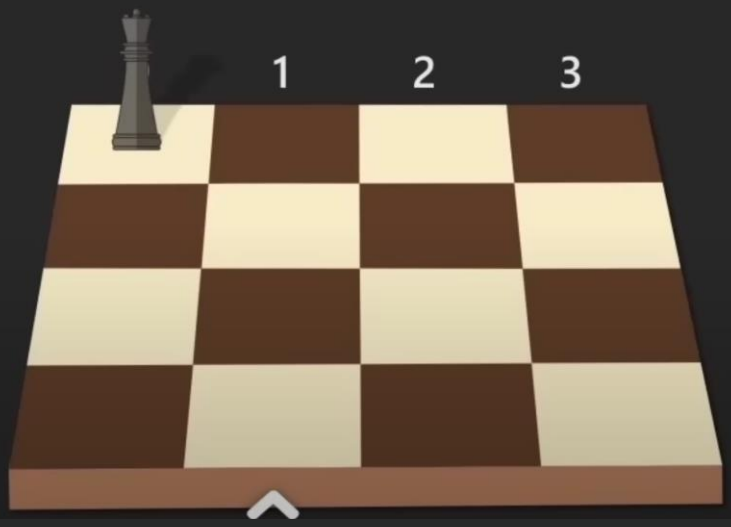
Problema reginei N este un puzzle matematic și un exemplu frecvent utilizat pentru a ilustra tehnica de backtracking. Scopul este de a plasa N regine pe o tablă de șah de dimensiune NxN astfel încât niciuna dintre ele să nu poată ataca celelalte. O regină poate ataca orice poziție de pe aceeași linie, coloană sau diagonală.

Algoritmul de backtracking este ales deoarece este eficient pentru a explora toate posibilitățile și pentru a reveni la decizii anterioare în cazul în care o soluție parțială nu poate fi extinsă la o soluție validă. În cazul reginei N, algoritmul începe cu o soluție parțială și încerca să o extindă treptat, încercând toate opțiunile valide la fiecare pas.

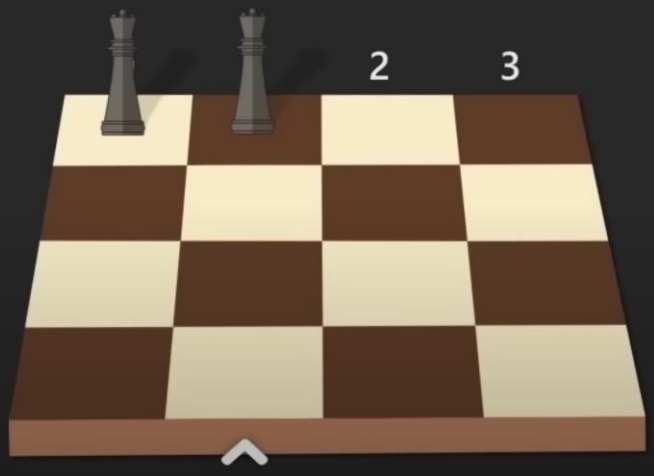
Explicația algoritmului în contextul problemei reginei N ar putea include următorii pași:

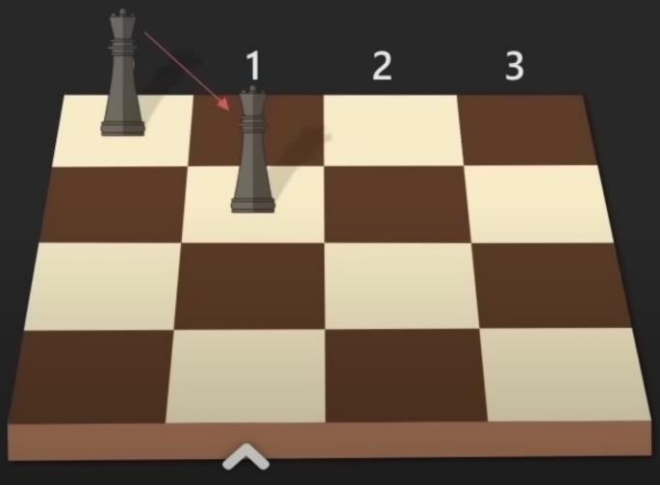
- Inițializarea unei table de șah goale de dimensiune NxN.
- Plasarea primei regine pe prima linie și prima coloană.
- Apelarea recursivă a algoritmului pentru a plasa celelalte regine pe celelalte linii, încercând fiecare poziție posibilă.
- Întoarcerea la o poziție anterioară (backtracking) atunci când nu este posibilă plasarea următoarei regine fără a ataca celelalte.
- Continuarea procesului până când toate reginele sunt plasate sau toate opțiunile au fost încercate.
- Această abordare oferă o soluție validă pentru problema reginei N și poate fi adaptată în funcție de cerințele specifice ale problemei.
- Implementarea poate varia în funcție de limbajul de programare utilizat și de detaliile algoritmului.

Let's start with placing queen at **row 0**.



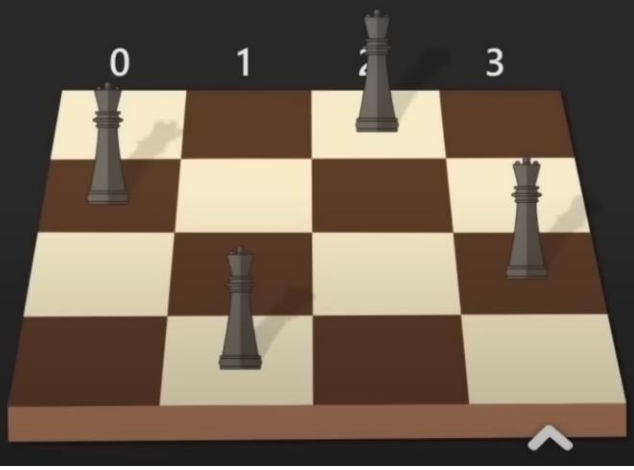
The queen is not safe here so we move to the next row.





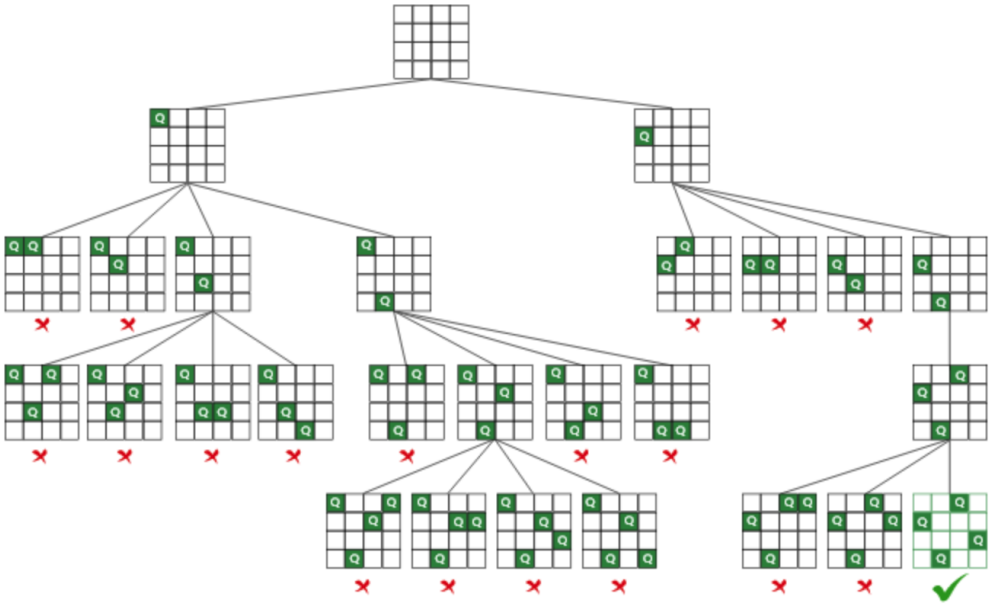
This position is safe. We move to the next **column 2** the same way and follow the same procedure.

Since all of the columns have a queen placed, we have our solution.



Cum functioneaza?

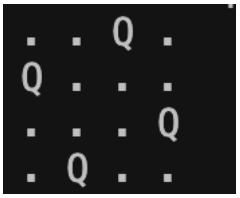
Exemplu pentru N=4

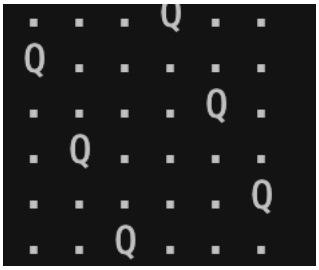


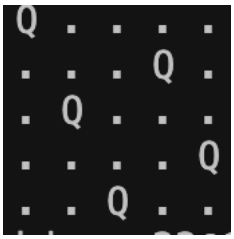
```
//Cerința pentru problema reginei N (N-Queens Problem) constă în a plasa N regine pe o
//tablă de șah de dimensiune NxN astfel încât nicio regină să nu poată ataca alte regine.
//Reginele se mișcă pe linii, coloane și diagonale și nu pot ocupa aceeași poziție cu o altă regină.


public class ProblemaRegineiN {
    final int N = 4;// N= 4-8, pot alege N de la 4 la 8

    void afiseazaSolutie(int tabla[][]) {
        for (int i = 0; i < N; i++) {
            for (int j = 0; j < N; j++) {
                if (tabla[i][j] == 1)
                    System.out.print("Q ");
                else
                    System.out.print(". ");
            }
            System.out.println();
        }
    }

    boolean esteSigur(int tabla[][], int rand, int coloana) {
        int i, j;

        // Verificăm această linie pe partea stângă
        for (i = 0; i < coloana; i++)
            if (tabla[rand][i] == 1)
                return false;

        // Verificăm diagonală superioară pe partea stângă
        for (i = rand, j = coloana; i >= 0 && j >= 0; i--, j--)
```

```
        if (tabla[i][j] == 1)
            return false;

    // Verificăm diagonală inferioară pe partea stângă
    for (i = rand, j = coloana; j >= 0 && i < N; i++, j--)
        if (tabla[i][j] == 1)
            return false;

    return true;
}

boolean rezolvaProblemaRegineiNUtil(int tabla[][], int coloana) {
    if (coloana >= N)
        return true;

    for (int i = 0; i < N; i++) {
        if (esteSigur(tabla, i, coloana)) {
            tabla[i][coloana] = 1;

            if (rezolvaProblemaRegineiNUtil(tabla, coloana + 1))
                return true;

            tabla[i][coloana] = 0;
        }
    }

    return false;
}

boolean rezolvaProblemaRegineiN() {
    int tabla[][] = new int[N][N];

    if (!rezolvaProblemaRegineiNUtil(tabla, 0)) {
        System.out.print("Solutie nu exista");
        return false;
    }

    afiseazaSolutie(tabla);
    return true;
}

public static void main(String args[]) {
    ProblemaRegineiN regina = new ProblemaRegineiN();
    regina.rezolvaProblemaRegineiN();
}
}
```

AFISARE

Pentru N=4 

Pentru N=6 

Pentru N=5 

Pentru N=7