

Backtrack

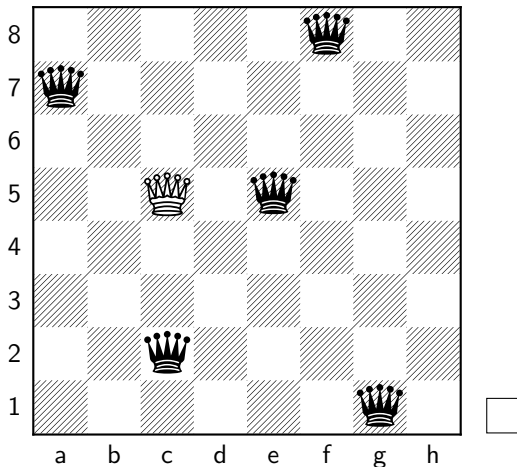
Giuseppe Persiano

Università di Salerno

Novembre, 2021

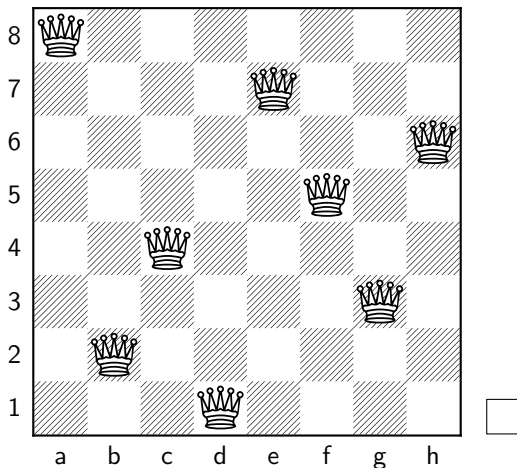
N regine

Piazzare 8 regine su una scacchiera in modo tale che nessuna regina attacchi un'altra regina.



N regine

Piazzare 8 regine su una scacchiera in modo tale che nessuna regina attacchi un'altra regina.



Algoritmo di backtrack

La filosofia del backtrack

- prova tutte le scelte che non sono in conflitto con quelle precedenti
- se non hai alternative, torna indietro a cambia una delle scelte precedenti

8 righe: $0, 1, \dots, 7$

Backtrack applicato al problema delle 8 regine

Prova a piazzare una regina sulla riga R

Regine sulle righe $0, 1, 2, \dots, R - 1$

- se $R = 8$
termina con successo
 - se $R < 8$
per ogni possibile posizione C sulla riga R
 - ▶ se posizione (R, C) non è sotto attacco
 - ★ **annota** dove hai messo la regina e passa ricorsivamente alla riga $R + 1$
 - ★ se la chiamata ricorsiva ha successo, termina con successo
(torna alla chiamata precedente alla riga $R - 1$)
- se hai provato tutte le posizioni, termina con insuccesso
(torna alla chiamata precedente alla riga $R - 1$)

prima chiamata $R = 0$

```
def solve(R,regine,N):  
    if R==N:  
        return True  
    for C in range(N):  
        regine[R]=C  
        if checkQueen(regine,R):  
            if solve(R+1,regine,N):  
                return True  
    return False
```

Per risolvere il problema con $N = 8$

```
soluzione=[None]*8  
if solve(R=0,regine=soluzione,N=8):  
    print(soluzione)
```

```

##controlla se la posizione della regina nella riga R
##e' compatibile con le regine nelle righe 0,1,...,R-1
def checkQueen(posizioni,R):
    r1=R
    c1=posizioni[R]
    for r in range(R):
        r2=r
        c2=posizioni[r]
        if attackingQ(r1,c1,r2,c2):
            return False
    return True

def attackingQ(r1,c1,r2,c2):
    return samecolumn(r1,c1,r2,c2) or sameMajorD(r1,c1,r2,c2) \
        or sameMinorD(r1,c1,r2,c2)

```

Controllare se due regine si attaccano

Regine a (r_1, c_1) e (r_2, c_2)

- controlla se sono sulla stessa riga:

Controllare se due regine si attaccano

Regine a (r_1, c_1) e (r_2, c_2)

- controlla se sono sulla stessa riga:
 - ▶ $r_1 == r_2$

Controllare se due regine si attaccano

Regine a (r_1, c_1) e (r_2, c_2)

- controlla se sono sulla stessa riga:
 - ▶ $r_1 == r_2$
- controlla se sono sulla stessa colonna:

Controllare se due regine si attaccano

Regine a (r_1, c_1) e (r_2, c_2)

- controlla se sono sulla stessa riga:
 - ▶ $r_1 == r_2$
- controlla se sono sulla stessa colonna:
 - ▶ $c_1 == c_2$

Controllare se due regine si attaccano

Regine a (r_1, c_1) e (r_2, c_2)

- controlla se sono sulla stessa riga:
 - ▶ $r_1 == r_2$
- controlla se sono sulla stessa colonna:
 - ▶ $c_1 == c_2$
- controlla se sono sulla stessa diagonale maggiore:

Controllare se due regine si attaccano

Regine a (r_1, c_1) e (r_2, c_2)

- controlla se sono sulla stessa riga:
 - ▶ $r_1 == r_2$
- controlla se sono sulla stessa colonna:
 - ▶ $c_1 == c_2$
- controlla se sono sulla stessa diagonale maggiore:
 - ▶ $c_1 - r_1 == c_2 - r_2$

Controllare se due regine si attaccano

Regine a (r_1, c_1) e (r_2, c_2)

- controlla se sono sulla stessa riga:
 - ▶ $r_1 == r_2$
- controlla se sono sulla stessa colonna:
 - ▶ $c_1 == c_2$
- controlla se sono sulla stessa diagonale maggiore:
 - ▶ $c_1 - r_1 == c_2 - r_2$
- controlla se sono sulla stessa diagonale minore:

Controllare se due regine si attaccano

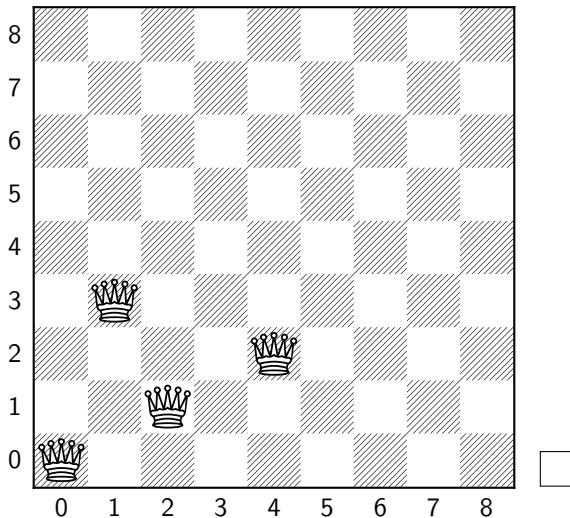
Regine a (r_1, c_1) e (r_2, c_2)

- controlla se sono sulla stessa riga:
 - ▶ $r_1 == r_2$
- controlla se sono sulla stessa colonna:
 - ▶ $c_1 == c_2$
- controlla se sono sulla stessa diagonale maggiore:
 - ▶ $c_1 - r_1 == c_2 - r_2$
- controlla se sono sulla stessa diagonale minore:
 - ▶ $c_1 + r_1 == c_2 + r_2$

R = 4

regine = [0, 2, 4, 1, None, None, None, None, None]

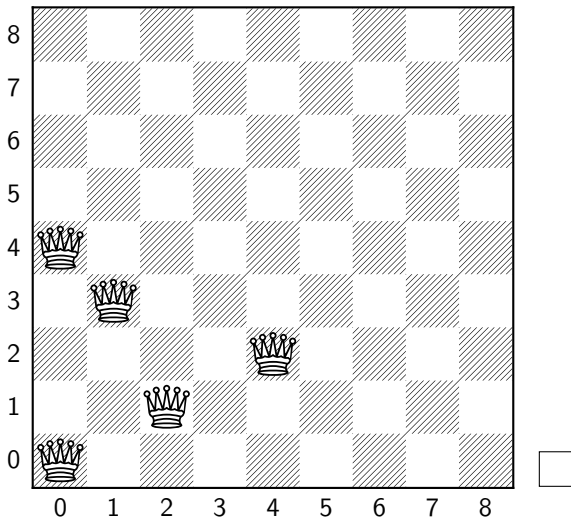
N = 9



R = 4
C = 0

regine = [0, 2, 4, 1, None, None, None, None]

N = 9

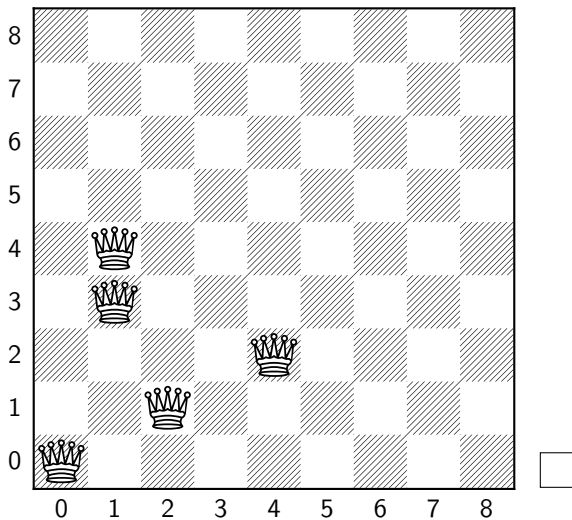


R = 4

regine = [0, 2, 4, 1, None, None, None, None]

N = 9

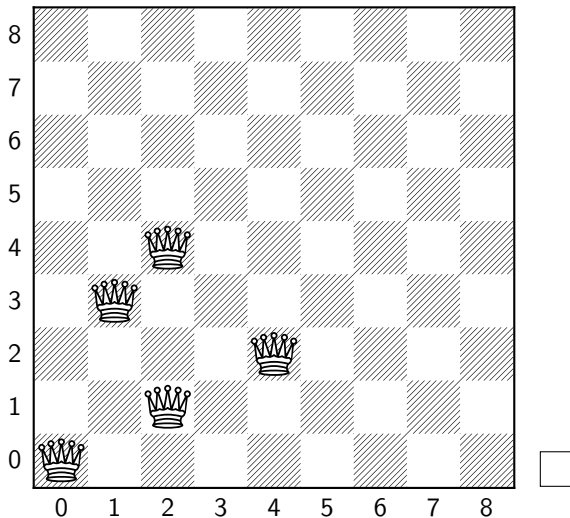
C=1



R = 4
C = 2

regine = [0, 2, 4, 1, None, None, None, None]

N = 9

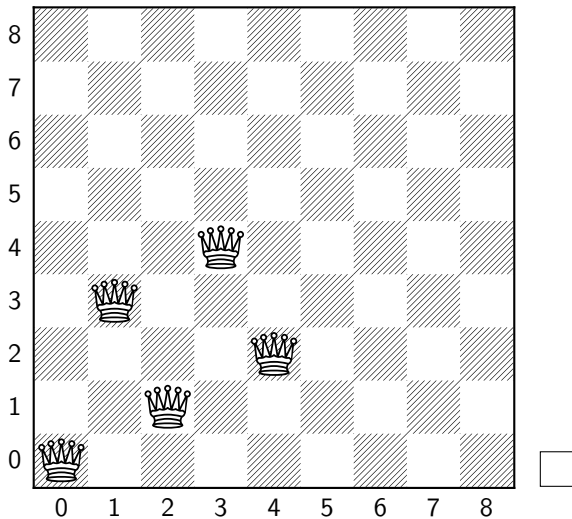


R = 4

regine = [0, 2, 4, 1, None, None, None, None]

N = 9

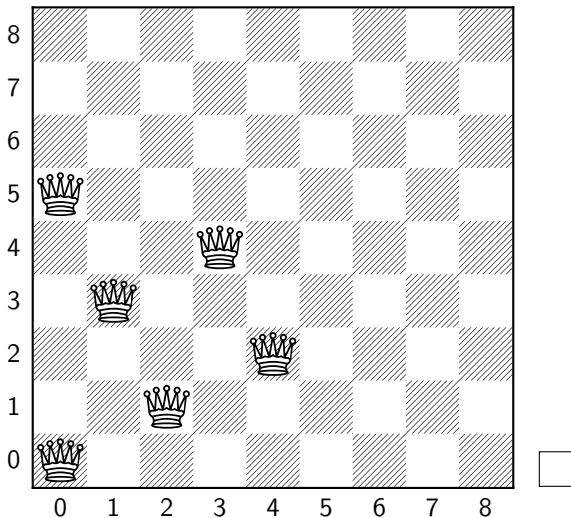
C=3



R = 5
C = 0

regine = [0, 2, 4, 1, 3, None, None, None, None]

N = 9

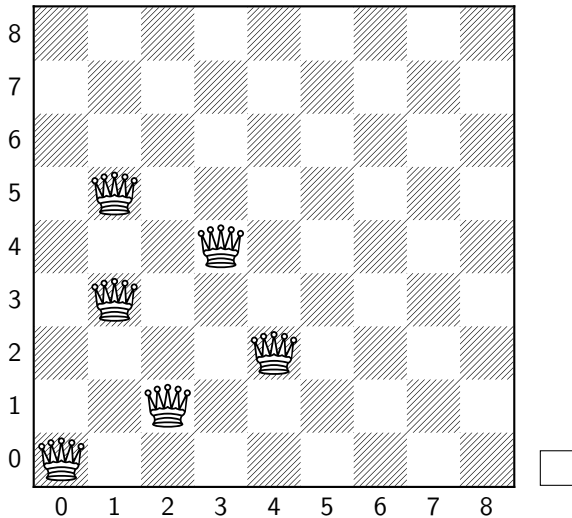


R = 5

regine = [0, 2, 4, 1, 3, None, None, None, None]

N = 9

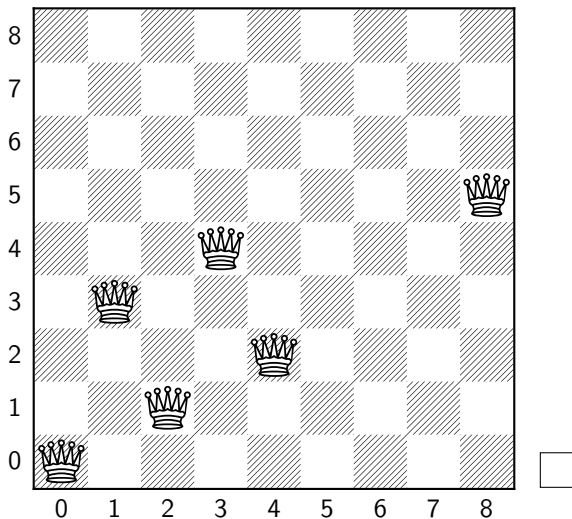
C=1



R = 5
C = 8

regine = [0, 2, 4, 1, 3, None, None, None]

N = 9

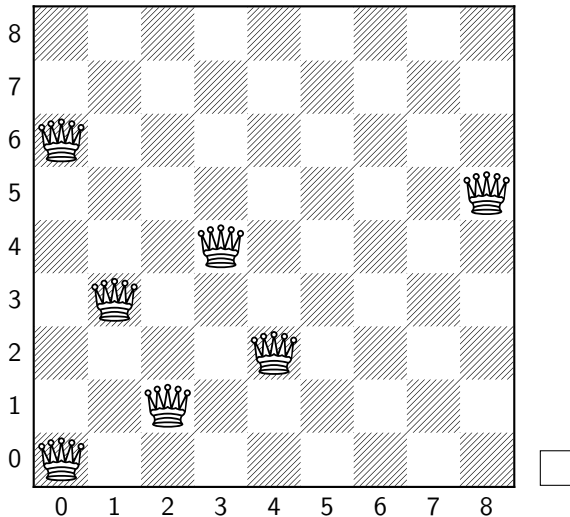


R = 6

regine = [0, 2, 4, 1, 3, 8, None, None, None]

N = 9

C=0

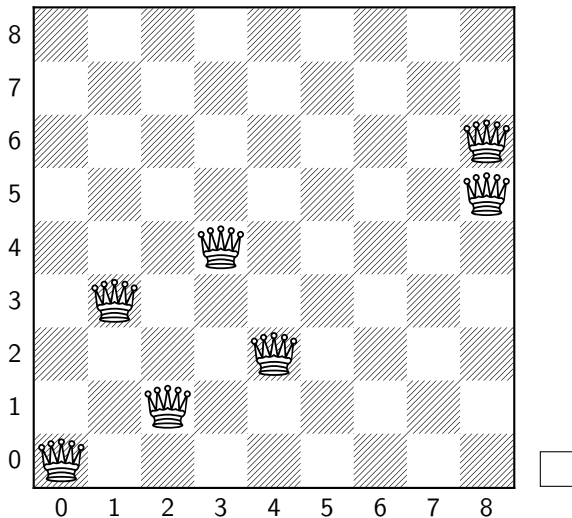


R = 6

regine = [0, 2, 4, 1, 3, 8, None, None, None]

N = 9

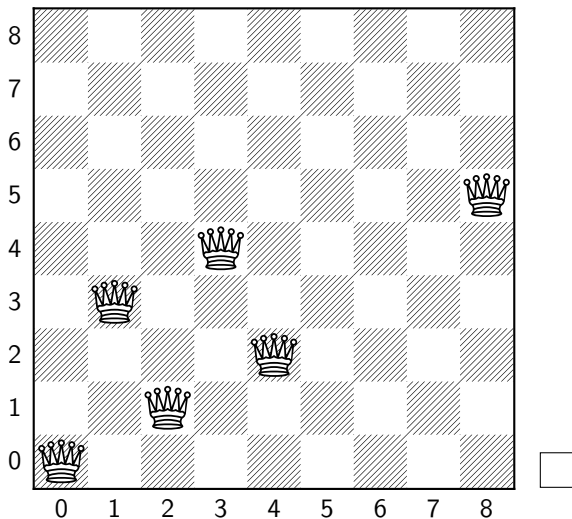
C=8



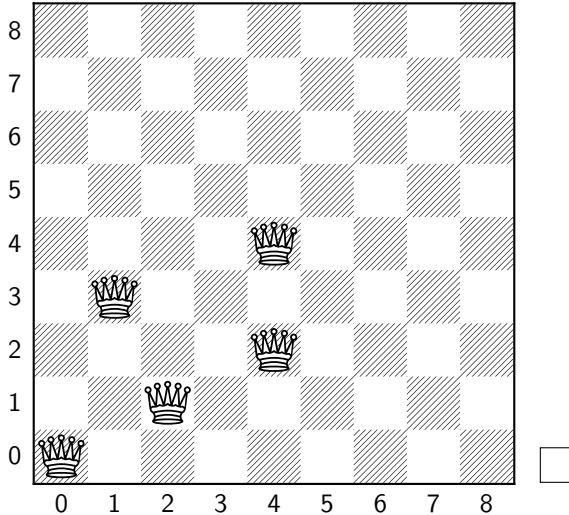
R = 5
C = 8

regine = [0, 2, 4, 1, 3, None, None, None]

N = 9



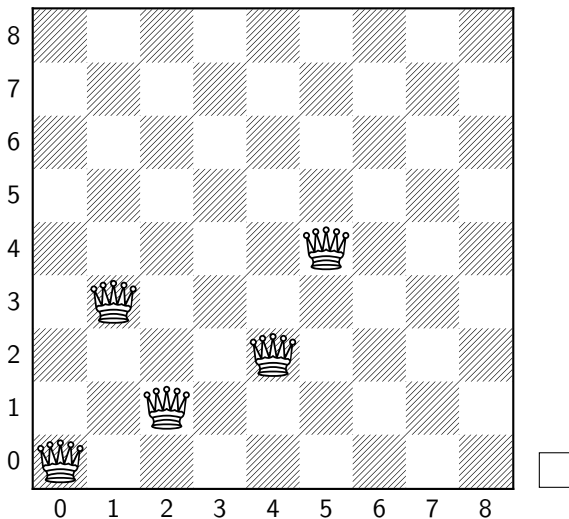
$R = 4$ $\text{regine} = [0, 2, 4, 1, \text{None}, \text{None}, \text{None}, \text{None}, \text{None}]$ $N = 9$
 $C=3 \Rightarrow C=4$



R = 4
C = 5

regine = [0, 2, 4, 1, None, None, None, None]

N = 9

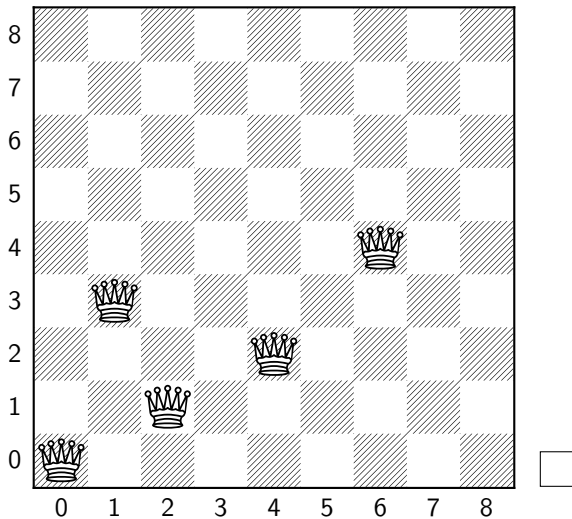


R = 4

regine = [0, 2, 4, 1, None, None, None, None]

N = 9

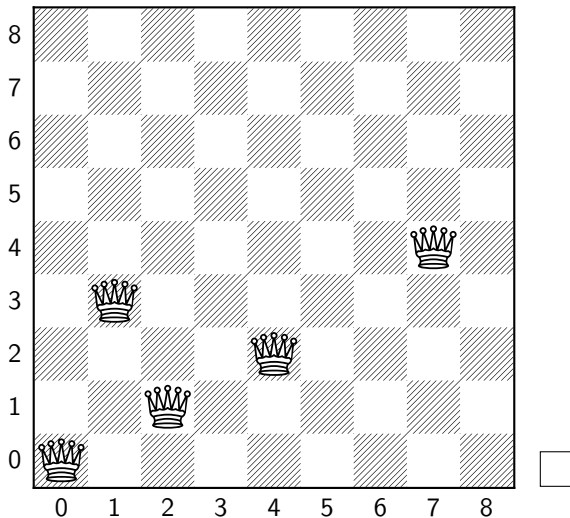
C=6



R = 4
C = 7

regine = [0, 2, 4, 1, None, None, None, None]

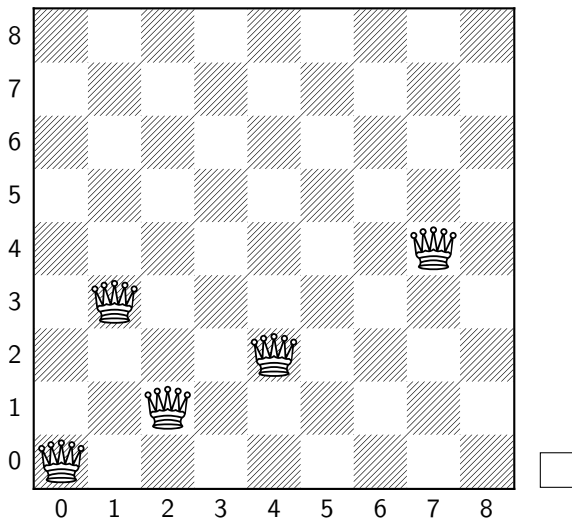
N = 9



R = 5

regine = [0, 2, 4, 1, 7, None, None, None]

N = 9



R = 5

regine = [0, 2, 4, 1, 7, None, None, None]

N = 9

C=0

