Licență informatică

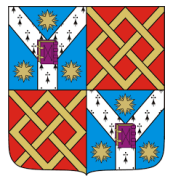
Problema Reginelor

A. Student Iacob Iustina

Facultatea de Matematică
Specializarea Matematică - Informatică

Universitatea "Al. I. Cuza" Iași
Facultatea de Matematică

Iulie 2022

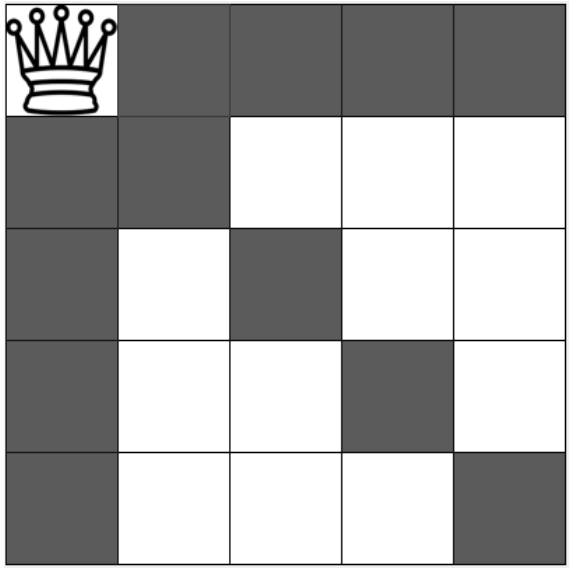


Cuprins

Problema reginelor

Enunț

Dându-se o tablă de șah de dimensiune nxn ($n > 1$) să se aranjeze pe ea n regine fără ca ele să se atace. Reamintim că o regină atacă linia, coloana și cele 2 diagonale pe care se află. În figura de mai jos celulele colorate mai închis sunt atacate de regina poziționată în caseta (0, 0).



Crearea ferestrei și a tabelului

Descriere

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris.

Listing Slider creation

```
1: window = Window("Queen", 800, 600)
2:
3: table = Table(window, 5, center=Point(400, 300))
```

Punctul de pornire al algoritmului

Listing Start Function

```
 1: def start():
 2:     slider_rows.set_visible(False)
 3:     start_button.set_visible(False)
 4:     w = table.squares[0][0].width
 5:     for i in range(table.rows):
 6:         queens.append({
 7:         'col': None,
 8:         'image': Image(window, width=w, height=w,
 9:                        path="../images/queen.png"),
10:         })
11:         queens[i]['image'].set_visible(False)
12:
13:     backtracking_thread = threading.Thread(
14:         target=backtracking_init)
15:     backtracking_thread.start()
```

Algoritmul

Explicație 1

Lorem ipsum dolor sit amet, consectetuer adipiscing elit.

Explicație 2

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis.

Explicație 3

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus.

Algoritmul

Explicație

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis.

```
1: def back(k):
2:     for i in range(table.rows):
3:         global is_running
4:         if not is_running:
5:             return False
6:         queens[k]['col'] = i
7:         queens[k]['image'].center =
8:                 table.squares[k][i].center
9:         queens[k]['image'].set_visible(True)
```

Algoritmul

```
1:              pause()
2:              if valid(k):
3:                  if solution(k):
4:                      show_solution()
5:                  else:
6:                      back(k + 1)
7:          queens[k]['image'].set_visible(False)
8:          pause()
```

Precizare

Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque.

Despre algoritm

Întrebare

Ce metodă utilizează acest algoritm?

Răspuns

Metoda utilizată este cea de backtracking, întrucât ... Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas.

Despre algoritm

Cazul $n \leq 3$

În această situație programul nu va afișa nici o variantă corectă, deoarece e imposibil să plasăm, de exemplu, 3 regine pe o tablă de dimensiune 3x3 fără să se atace reciproc.

Cazul $n > 3$

În această situație, programul va afișa mereu o variantă corectă.

Întrebări

Întrebarea 1

Ce complexitate are algoritmul?

Întrebarea 2

Este cea mai eficientă variantă?

Întrebarea 3

Care sunt avantajele utilizării acestui algoritm realizat prin metoda backtracking?

Bibliografie I

[jav11] [Kar15] [Slo13]


javaTpoint, *N-Queens Problem*, `https://www.javatpoint.com/n-queens-problems`, 2011, [Online; accessed 19-July-2022].


Narasimha Karumanchi, *Algorithm design techniques: Recursion, backtracking, greedy, divide and conquer, and dynamic programming*, Narasimha Karumanchi, 2015.


Leila Sloman, *Mathematician answers chess problem about attacking queens*, Quantamagazine **2** (2013), no. 7.