

Compléments

DIU Enseignement de l'informatique au lycée
Bloc 5 : Algorithmique avancée

Bruno Grenet – Université de Montpellier

Juin-juillet 2020

Algorithme A*

OpenStreetMap

Traces GPS · Journaux des utilisateurs · Droits d'auteur · Aide · À propos

Arad, Roumanie
Bucarest, 010051, Roumanie
En voiture (OSRM)
Inverser les directions

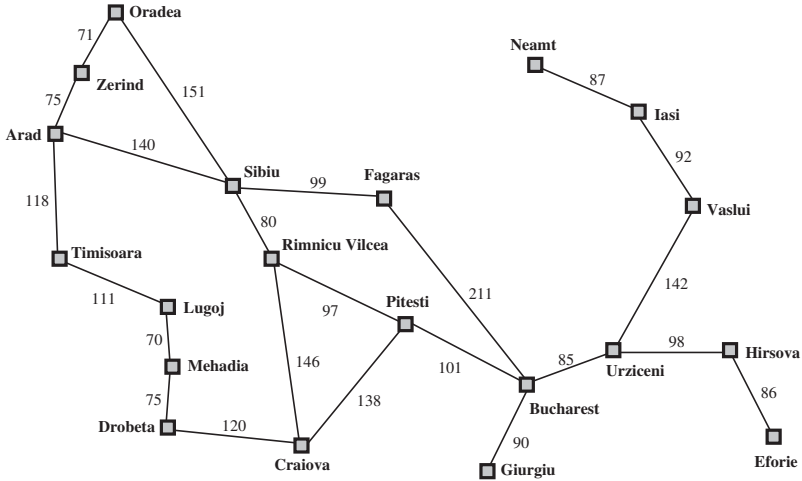
Itinéraire

Distance: 576km. Temps: 7:36.

1. Démarrez à **Strada General Vasile Milea**
2. Tournez à gauche sur **Strada Romul Veliclu**
3. Tournez à gauche sur **Bulevardul Revoluției (DN79)**
4. Tournez à gauche sur **Strada Xenopol**
5. À la bifurcation, tournez à droite sur **Bulevardul Decebal**
6. Au rond-point prendre la sortie 2nd sur **Calea Romanilor (DN69)**
7. Sortir du rond-point vers **Calea Romanilor (DN69)**
8. Continuer sur **Podul Traian (DN69)**
9. Continuer sur **Strada Banatului (DN69)**
10. Continuer sur **Calea Timișorii (DN69)**
11. Au rond-point prendre la sortie 1er sur **Calea Timișorii (DN69)**
12. Sortir du rond-point vers **Calea Timișorii (DN69)**
13. Au rond-point prendre la sortie 2nd sur **DN69**
14. Sortir du rond-point vers **DN69**
15. Au rond-point prendre la sortie 1er sur **voie sans nom**

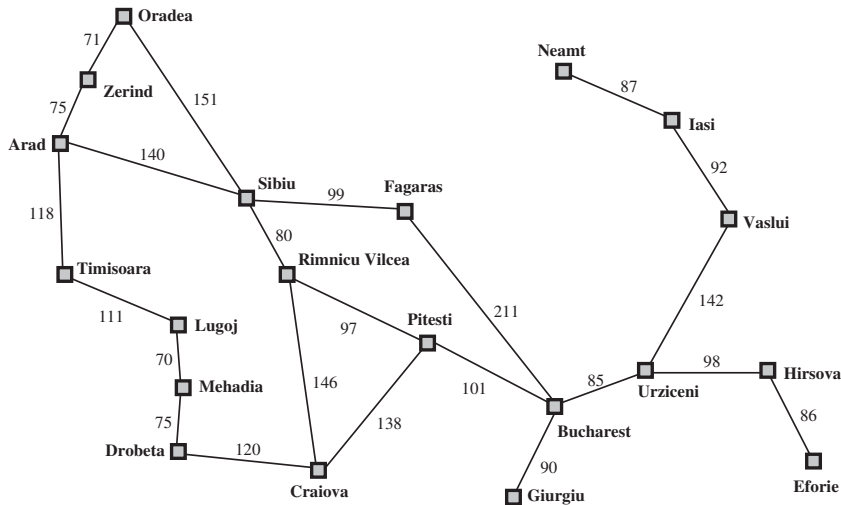
© Contributors of OpenStreetMap · Faire un don, Conditions du site web et de l'API

Avec l'algorithme de Dijkstra



S. Russel, S. Norvig. *Artificial Intelligence: A Modern Approach*, 3rd ed., Prentice Hall, 2010

Avec l'algorithmme A*



Arad	366
Bucharest	0
Craiova	160
Drobeta	242
Eforie	161
Fagaras	176
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadia	241
Neamt	234
Oradea	380
Pitesti	100
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

Caractéristiques de l'algorithme

Si l'heuristique est *cohérente* (inégalité triangulaire), l'algorithme A^* est correct et optimal.

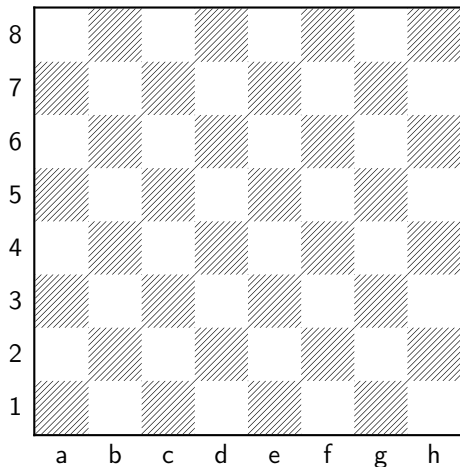
Caractéristiques de l'algorithme

Si l'heuristique est *cohérente* (inégalité triangulaire), l'algorithme A^* est correct et optimal.

- ▶ Utilisé pour les cartes routières
- ▶ Autres domaines, autres heuristiques

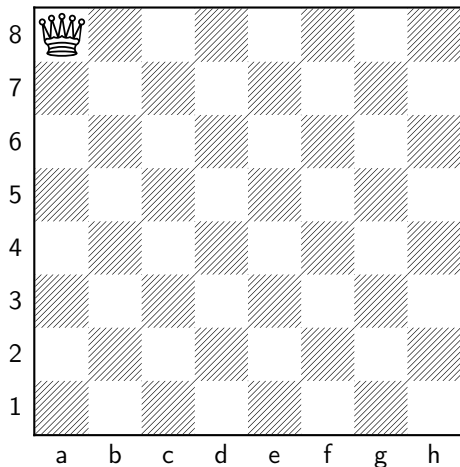
Le problème des huit reines

Peut-on placer 8 reines sur un échiquier, sans qu'elles s'attaquent l'une l'autre ?



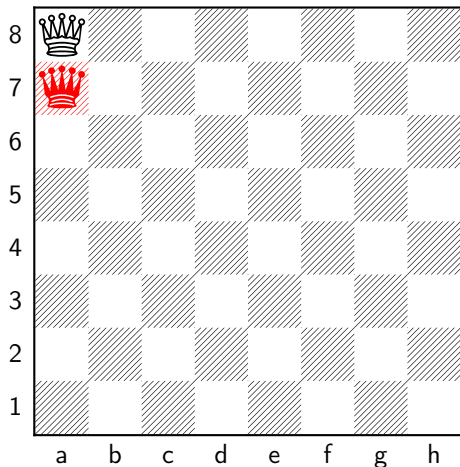
Le problème des huit reines

Peut-on placer 8 reines sur un échiquier, sans qu'elles s'attaquent l'une l'autre ?



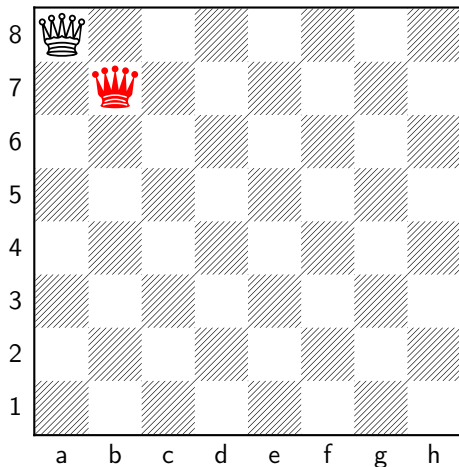
Le problème des huit reines

Peut-on placer 8 reines sur un échiquier, sans qu'elles s'attaquent l'une l'autre ?



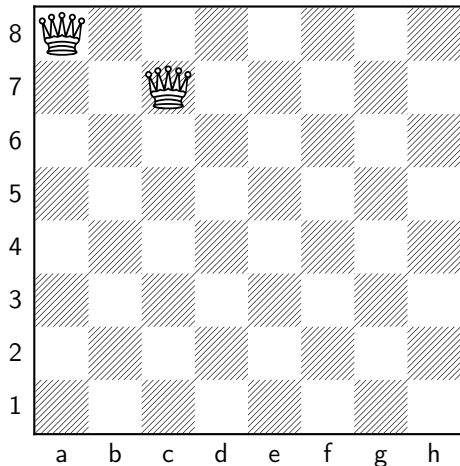
Le problème des huit reines

Peut-on placer 8 reines sur un échiquier, sans qu'elles s'attaquent l'une l'autre ?



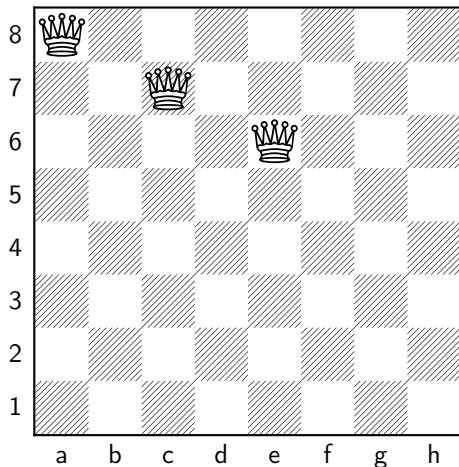
Le problème des huit reines

Peut-on placer 8 reines sur un échiquier, sans qu'elles s'attaquent l'une l'autre ?



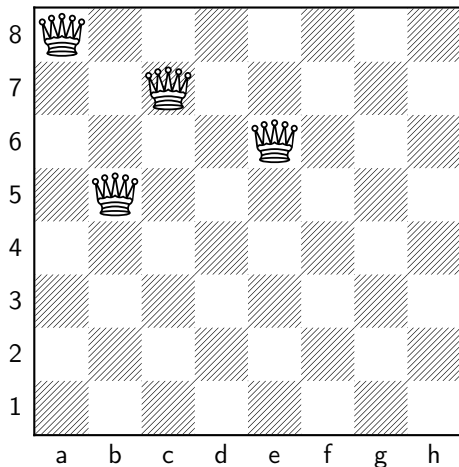
Le problème des huit reines

Peut-on placer 8 reines sur un échiquier, sans qu'elles s'attaquent l'une l'autre ?



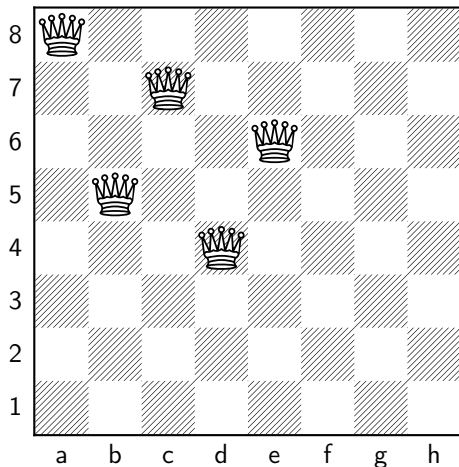
Le problème des huit reines

Peut-on placer 8 reines sur un échiquier, sans qu'elles s'attaquent l'une l'autre ?



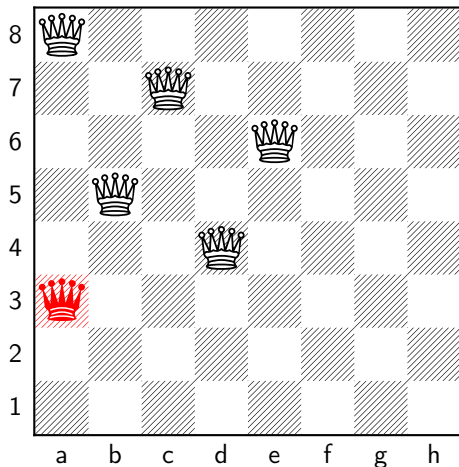
Le problème des huit reines

Peut-on placer 8 reines sur un échiquier, sans qu'elles s'attaquent l'une l'autre ?



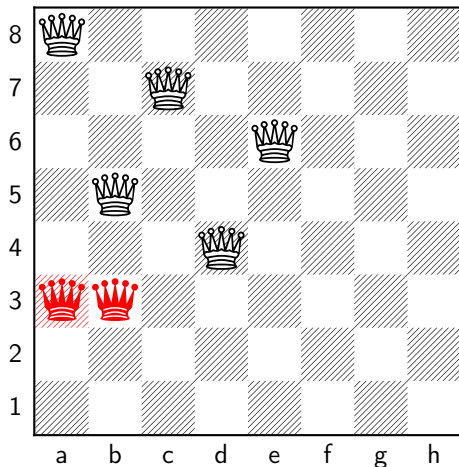
Le problème des huit reines

Peut-on placer 8 reines sur un échiquier, sans qu'elles s'attaquent l'une l'autre ?



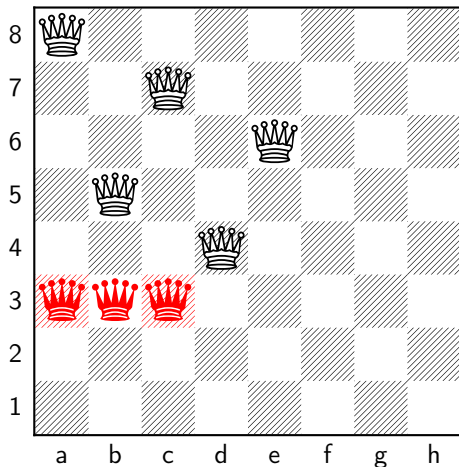
Le problème des huit reines

Peut-on placer 8 reines sur un échiquier, sans qu'elles s'attaquent l'une l'autre ?



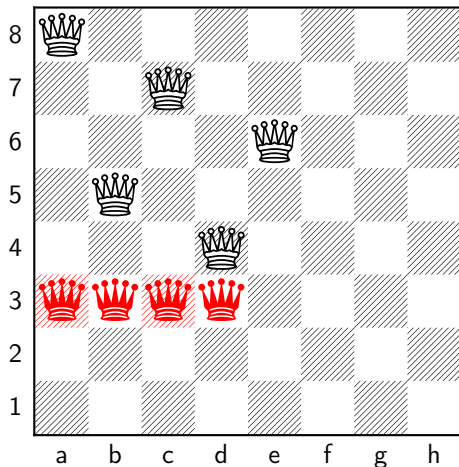
Le problème des huit reines

Peut-on placer 8 reines sur un échiquier, sans qu'elles s'attaquent l'une l'autre ?



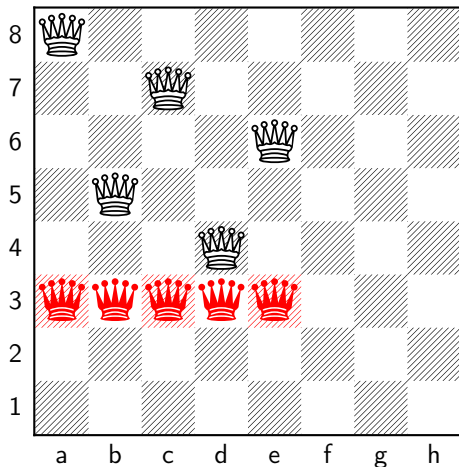
Le problème des huit reines

Peut-on placer 8 reines sur un échiquier, sans qu'elles s'attaquent l'une l'autre ?



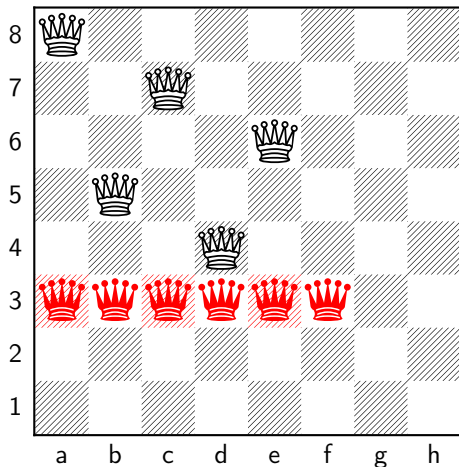
Le problème des huit reines

Peut-on placer 8 reines sur un échiquier, sans qu'elles s'attaquent l'une l'autre ?



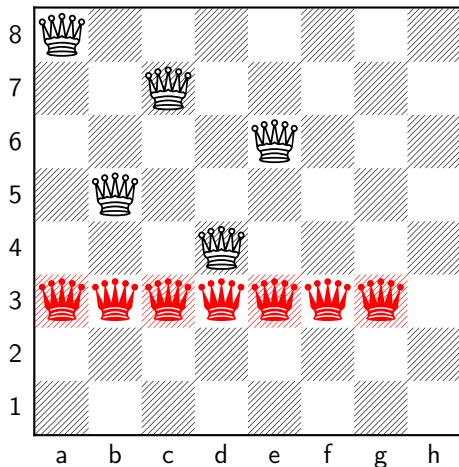
Le problème des huit reines

Peut-on placer 8 reines sur un échiquier, sans qu'elles s'attaquent l'une l'autre ?



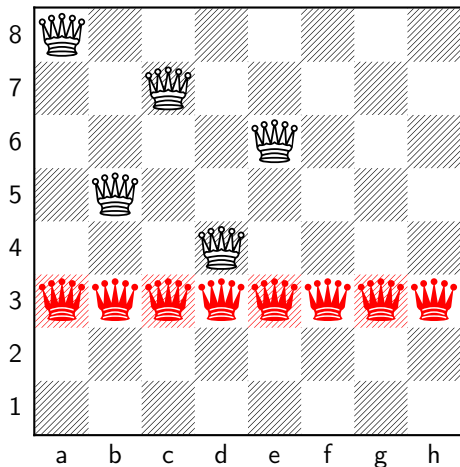
Le problème des huit reines

Peut-on placer 8 reines sur un échiquier, sans qu'elles s'attaquent l'une l'autre ?



Le problème des huit reines

Peut-on placer 8 reines sur un échiquier, sans qu'elles s'attaquent l'une l'autre ?



Quels algorithmes ?

- ▶ Force brute : OK pour $n = 8$, mais après ?
- ▶ *Backtrack*, *Branch-and-bound*, etc.
- ▶ Programmation par contraintes
- ▶ Programmation logique (SAT solveurs par ex.)
- ▶ Algorithme génétique
- ▶ ...

Quels algorithmes ?

- ▶ Force brute : OK pour $n = 8$, mais après ?
- ▶ *Backtrack*, *Branch-and-bound*, etc.
- ▶ Programmation par contraintes
- ▶ Programmation logique (SAT solveurs par ex.)
- ▶ Algorithme génétique
- ▶ ...

Algorithmes \pm efficaces, mais complexes

Quels algorithmes ?

- ▶ Force brute : OK pour $n = 8$, mais après ?
- ▶ *Backtrack*, *Branch-and-bound*, etc.
- ▶ Programmation par contraintes
- ▶ Programmation logique (SAT solveurs par ex.)
- ▶ Algorithme génétique
- ▶ ...

Algorithmes \pm efficaces, mais complexes

Algorithme probabiliste

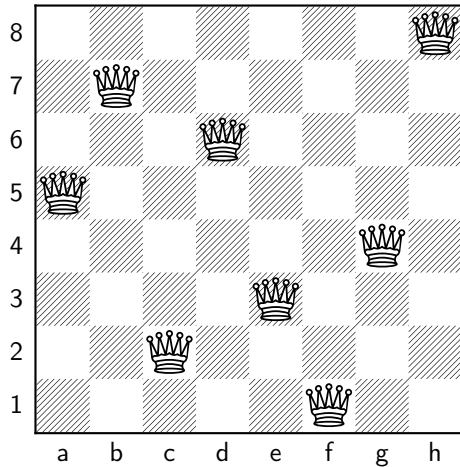
À chaque ligne :

- ▶ si aucune case disponible : recommencer à zéro
- ▶ choisir une case disponible aléatoire
- ▶ propager les contraintes

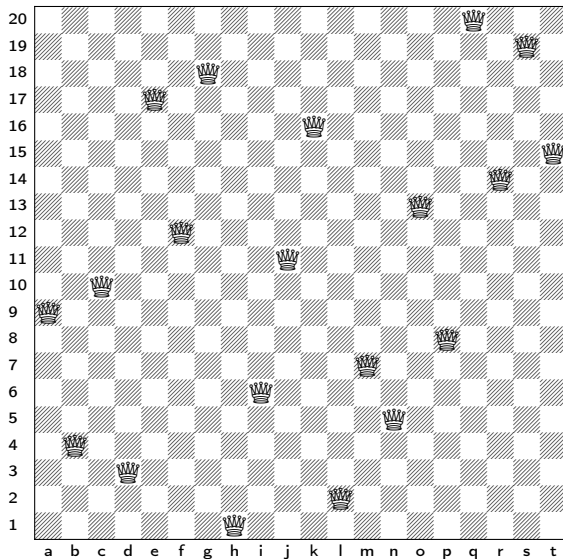
Le code complet

```
from random import choice
def huitreines(n = 8):
    E = [[0]*n for _ in range(n)]
    for i in range(n):
        dispo = [j for j in range(n) if E[i][j] != -1]
        if dispo == []:
            return huitreines(n)
        j = choice(dispo)
        E[i][j] = 1
        for l in range(i+1, n):
            E[l][j] = -1
            if j+l-i < n: E[l][j+l-i] = -1
            if j-l+i >=0: E[l][j-l+i] = -1
    return [[max(0,x) for x in ligne] for ligne in E]
```

Résultats



Résultats



Algorithmique probabiliste

Un algorithme *probabiliste* utilise des *bits aléatoires*.

Algorithmique probabiliste

Un algorithme *probabiliste* utilise des *bits aléatoires*.

- ▶ Heuristique aléatoire \neq algorithme probabiliste
 - ▶ Garantie des algos probabilistes
 - ▶ Heuristique aléatoire \simeq algo. probabiliste sans preuve

Algorithmique probabiliste

Un algorithme *probabiliste* utilise des *bits aléatoires*.

- ▶ Heuristique aléatoire \neq algorithme probabiliste
 - ▶ Garantie des algos probabilistes
 - ▶ Heuristique aléatoire \simeq algo. probabiliste sans preuve
- ▶ Bits aléatoires ?
 - ▶ Théorie : source extérieure d'aléa
 - ▶ Pratique :
 - ▶ générateurs de *pseudo-aléa*
 - ▶ rarement : source extérieure d'aléa (quantique, bruit ambiant, etc.)
 - ▶ Python : bibliothèque `random`

Conclusion

Deux thématiques (très) avancées

Conclusion

Deux thématiques (très) avancées

- ▶ Algorithmique probabiliste :
 - ▶ Qualité des générateurs pseudo-aléatoires
 - ▶ Développement d'algorithmes et preuves
 - ▶ Capacité théorique
 - ▶ Extension : algorithmique quantique

Conclusion

Deux thématiques (très) avancées

- ▶ Algorithmique probabiliste :
 - ▶ Qualité des générateurs pseudo-aléatoires
 - ▶ Développement d'algorithmes et preuves
 - ▶ Capacité théorique
 - ▶ Extension : algorithmique quantique
- ▶ Intelligence artificielle :
 - ▶ Domaine extrêmement large (algorithmique, représentation des connaissances, apprentissage, perception du monde réel, ...)
 - ▶ Accent mis sur l'utilité pratique...
 - ▶ ... mais beaucoup de théorie passionnante !
 - ▶ Assez éloigné de l'image *grand public*