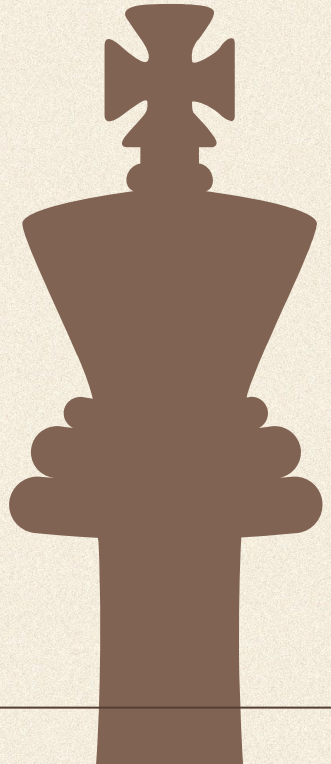
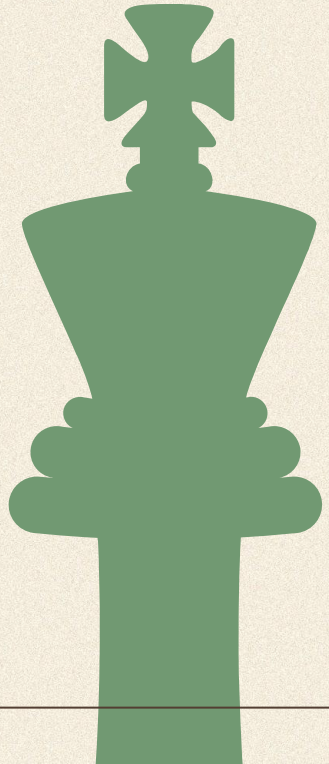


Problème des n reines

Compte-rendu d'une comparaison d'algorithmes



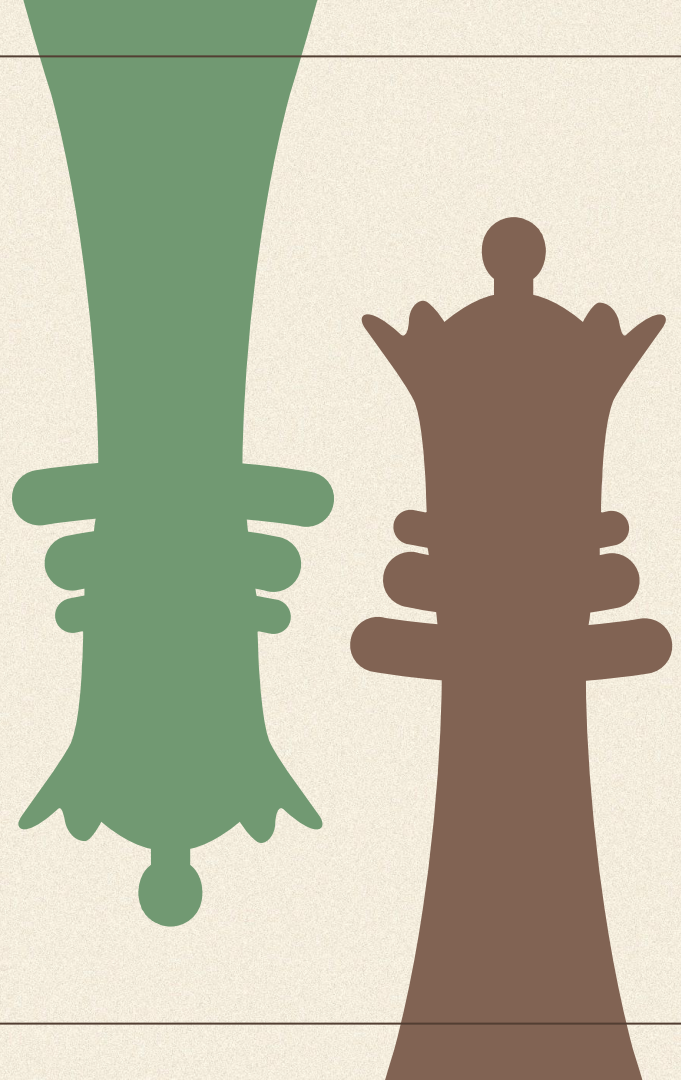
SOMMAIRE

01 Introduction

02 L'Algorithme

03 Comparaison

04 Conclusion



01

Introduction

Le problème des n reines

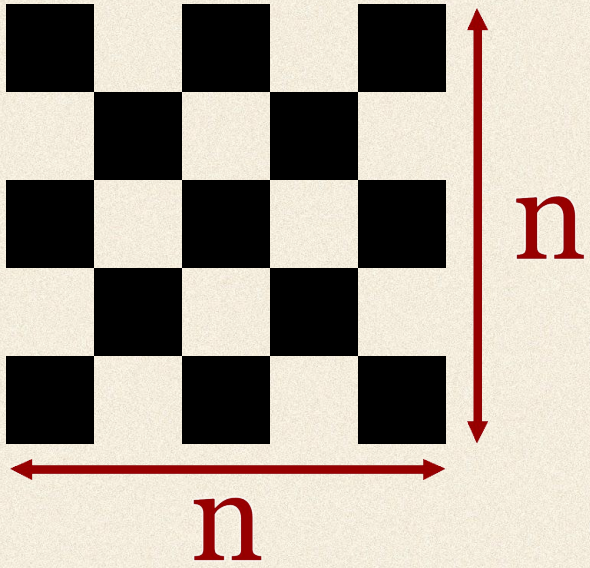
Problème mathématique de 1848

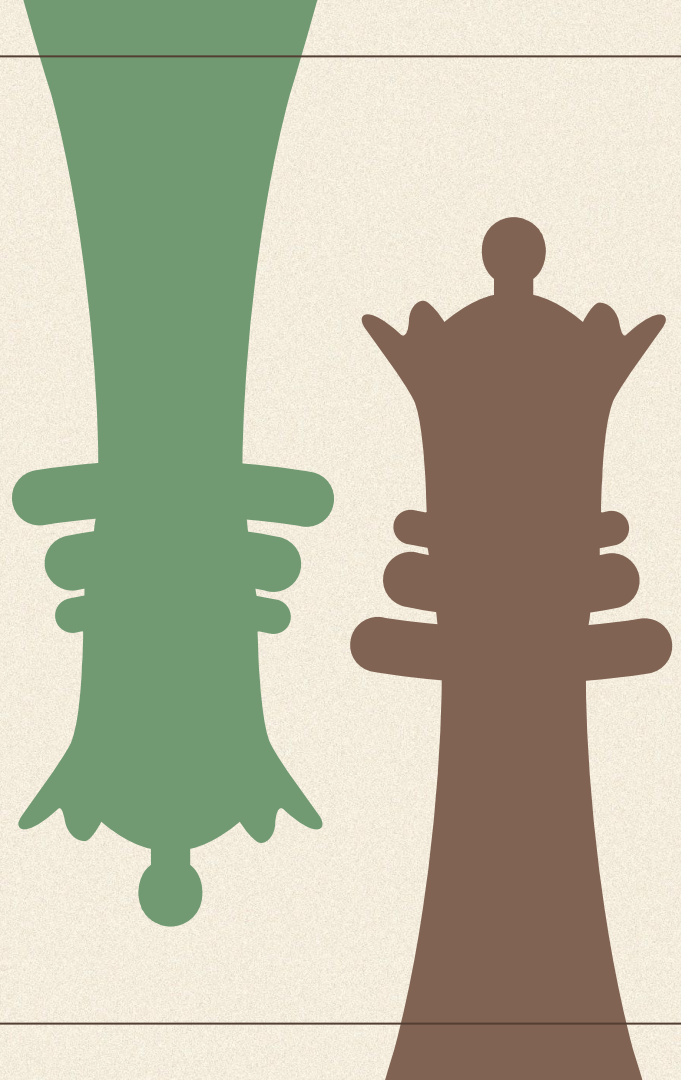
On place n reines sur un échiquier de taille n × n

Aucune reines ne doit en menacer une autre

n × ♛

♛ ♛ ♛ ♛ ♛





02

L’Algorithme

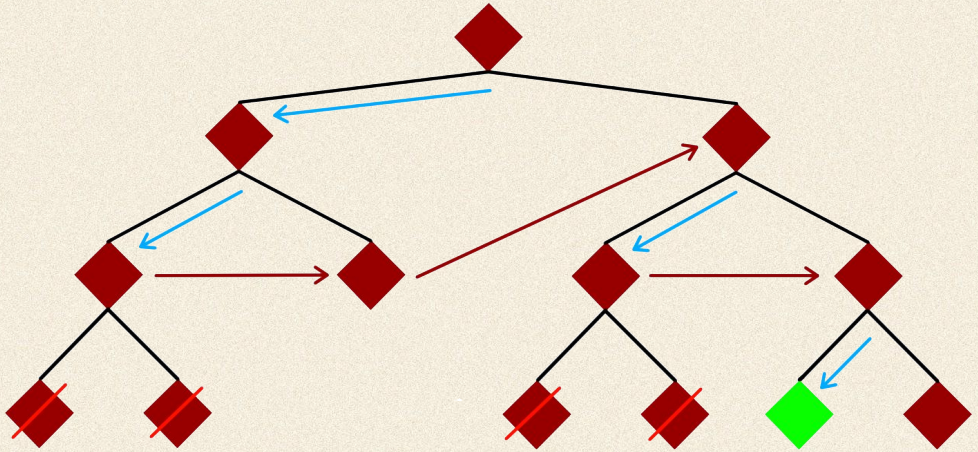
Algorithme de backtracking

Explore toutes les **configurations** valides de l'échiquier.

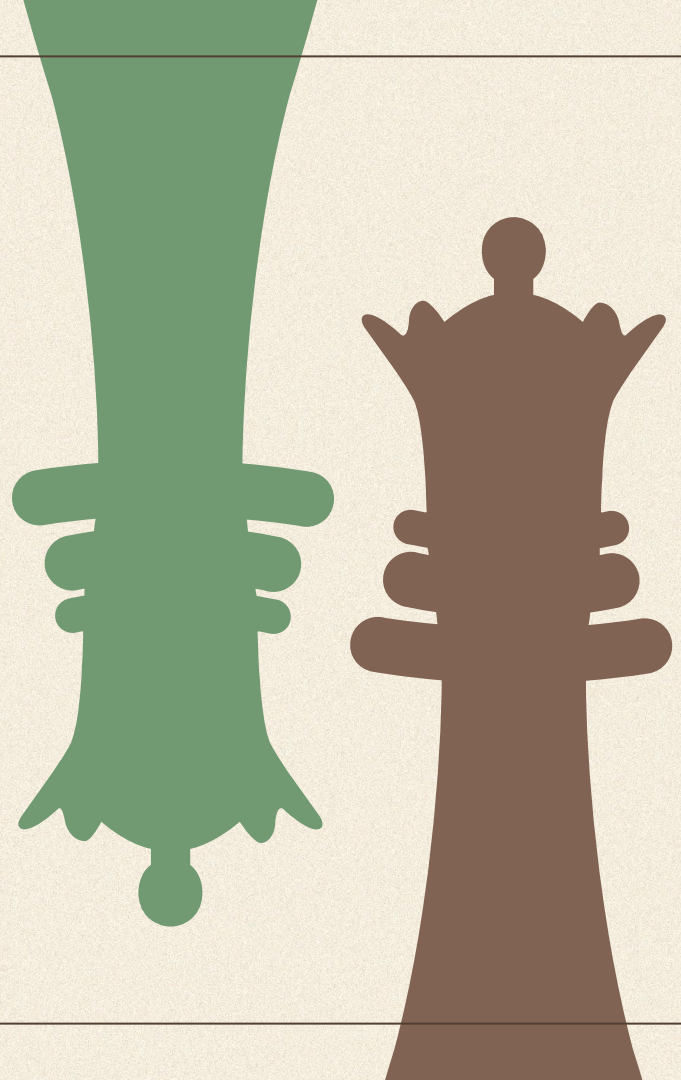
En cas de configuration invalide, **retour en arrière**

Efficace pour de **petites valeurs** de n.

La complexité augmente de manière **exponentielle** avec n.



Exemple d'exécution d'un algorithme de backtracking



03

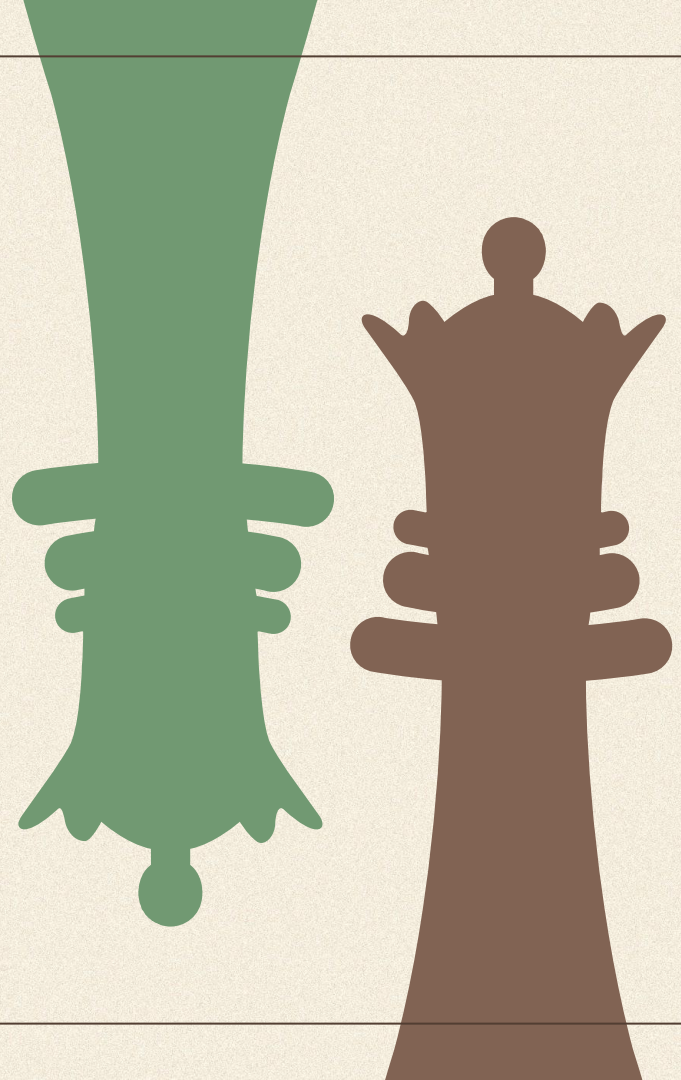
Comparaison

Comparaison

L'algorithme de **force brute** teste **toutes les configurations** d'échiquier possibles et garde les configurations valides

L'algorithme de **recherche locale** utilise une solution aléatoire modifiée pour trouver rapidement une solution satisfaisante

L'algorithme de **Backtracking** ou **Retour sur trace** est simple à implémenter et efficient sauf pour les valeurs de n élevées. Les solutions peuvent être calculées en parallèle.



04

Conclusion

Conclusion

Plusieurs algorithmes

Backtracking : **Simple** mais **lent** pour n élevé

Recherche Locale : **Peu précis** mais **rapide**

Force brute : **Simple** mais très **lent**

Améliorations possibles

Algorithmes **hybrides** : combiner le **meilleur** de plusieurs algorithmes

Optimisation par les **fourmis**

Réseaux de **neurones**