

SAÉ 2.02

Exploration algorithmique



LE PROBLÈME DES

N-REINES

Introduction

Le problème de N-reines est un problème posé par le problémiste allemand Max Bezzel en 1848. Les premières solutions à ce problème ont été publiées en 1850 par Franz Nauck et depuis plusieurs mathématiciens se sont mis à la résolution de ce problème, y compris Carl Friedrich Gauss.

Il existe plusieurs types d'algorithmes traitant du problème des N reines pour un échiquier de côté N, visant à notamment :

- Obtenir une solution
- Obtenir toutes les solutions
- Obtenir le nombre de solutions existantes

Nous avons choisi de nous focaliser sur des algorithmes permettant d'obtenir une solution.

Deux algorithmes sont présentés dans ce document : l'algorithme du masque et l'algorithme de symétrie.

Ces algorithmes sont **déterministes**, ce qui signifie que pour les mêmes arguments en entrée, la solution en sortie sera toujours la même.

Cela peut ne pas être le cas lorsqu'un algorithme utilise l'aléatoire par exemple.

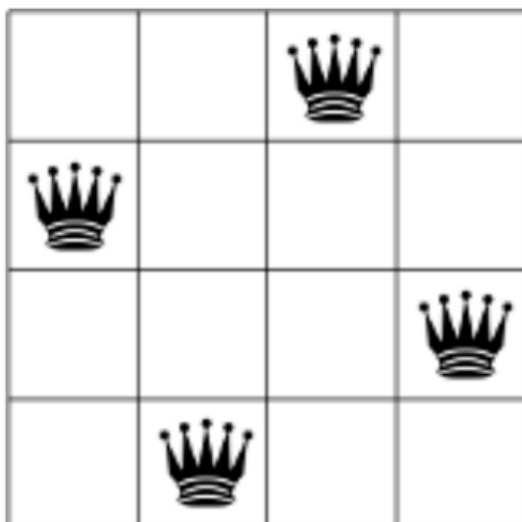


Fig. 1. exemple de solution au problème des N reines pour $N=4$

Masque

L'algorithme du masque est basé sur le principe du *backtracking*, qui consiste à parcourir récursivement les cases de l'échiquier et à placer une reine dès que possible. Quand on arrive dans un sans issue (c'est-à-dire qu'on ne peut plus progresser dans la résolution), on revient en arrière en enlevant la dernière reine. La résolution se fait par colonnes de l'échiquier. Pour chaque colonne, on place la reine sur une ligne.

Deux optimisations sont mises en place dans cet algorithme pour améliorer sa performance.

1. L'échiquier est représenté par une matrice carrée d'entiers naturels de côté N . Chaque valeur représente le nombre de reines menaçant la case. Les cases menacées par au moins une reine sont ignorées lors de la résolution (d'où le nom de "masque"), ce qui permet de gagner du temps en faisant moins de travail inutile.

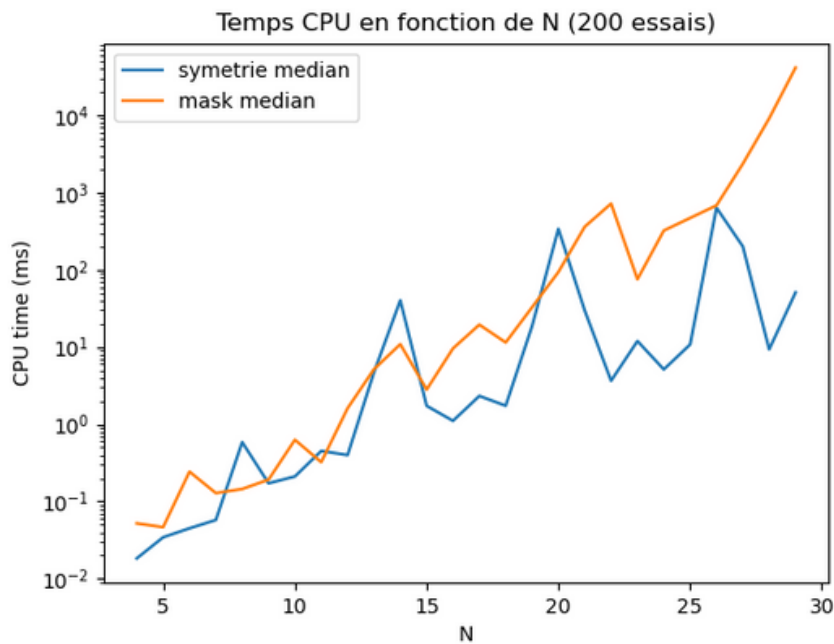
2. L'ordre de résolution des colonnes, au lieu d'être linéaire (de 0 à $N-1$), est basé sur un "tangage" de $N/2$ vers 0 et $N-1$. Cela signifie qu'on part de la colonne du milieu, puis qu'on s'éloigne progressivement vers les extrémités en alternant entre gauche et droite. Pour $N=6$, on obtient l'ordre suivant : 2, 3, 1, 4, 0, 5.

Cela a pour effet d'éliminer plus de possibilités au début de la résolution et donc de progresser plus rapidement.

Symétrie

L'algorithme de la symétrie est un algorithme de backtracking. Il cherche une solution spécifique où la position des reines est organisée de manière symétrique suivant l'axe central.

Pour cela, on pose les reines deux par deux ce qui va permettre de réduire grandement le temps d'exécution. Dans le cas où N est impair le programme ne pose de reine que dans la colonne centrale. Cependant, dans certains cas comme dans un échiquier de 32 par 32, la solution met beaucoup de temps (21,6 secondes). Or, dans la plupart des cas, une solution est trouvée très rapidement.



Faire la médiane sur 200 exécutions semblait être un bon compromis entre précision des résultats et temps d'exécution.

Fig. 2. Ce graphique se base sur le temps CPU, c'est-à-dire le temps passé par le processeur à exécuter l'algorithme, pour comparer la performance des deux algorithmes.

On voit qu'ils sont plutôt similaires. Cependant certaines valeurs de N (reines) prennent plus de temps à résoudre pour l'algorithme de symétrie (notamment 14, 20 et 26).

Graphique réalisé avec NumPy.

Conclusion

Nous avons développé différents algorithmes en Python pour trouver des solutions au problème des N-reines.

Ces algorithmes varient dans leur performance et dans leur fiabilité.

Cette SAÉ nous a permis d'améliorer notre connaissance de Python, ainsi que de perfectionner notre capacité à comprendre et à mettre au point des algorithmes.

Nous avons aussi pu renforcer notre cohésion lors du travail en équipe et notre organisation par l'utilisation de GitHub.