

## Devoir d'informatique numéro 1

---

Rendre le devoir sous la forme de un fichier par exercice, qui porte le nom

`DL1_[numero de l'exercice]_[NOM].py`

où [NOM] est en majuscule sans accents. Par exemple `DL1_2_CONDUCHÉ.py`.

### Exercice 1 (Tours de Hanoï)

Le problème des tours de Hanoï est un jeu de réflexion imaginé par le mathématicien français Édouard Lucas et consistant à déplacer des disques de diamètres différents d'une tour de « départ » à une tour d'« arrivée » en passant par une tour « intermédiaire » et ceci en un minimum de coups, tout en respectant les règles suivantes :

- on ne peut déplacer plus d'un disque à la fois ;
- on ne peut placer un disque que sur un autre disque plus grand que lui ou sur un emplacement vide.

Réalisation : Les trois piquets sont représentés par une liste de trois sous-listes : `piquets=[[...], [...], [...]]` où chaque sous-liste représente un des trois piquets. Les disques sont des entiers dont la valeur représente la taille du disque. Par exemple :



FIGURE 1 – Configuration initiale  
`piquets=[[6, 5, 4, 3, 2, 1], [], []]`



FIGURE 2 – Configuration finale  
`piquets=[[], [], [6, 5, 4, 3, 2, 1]]`

Voici une solution pour  $n = 3$  palets :

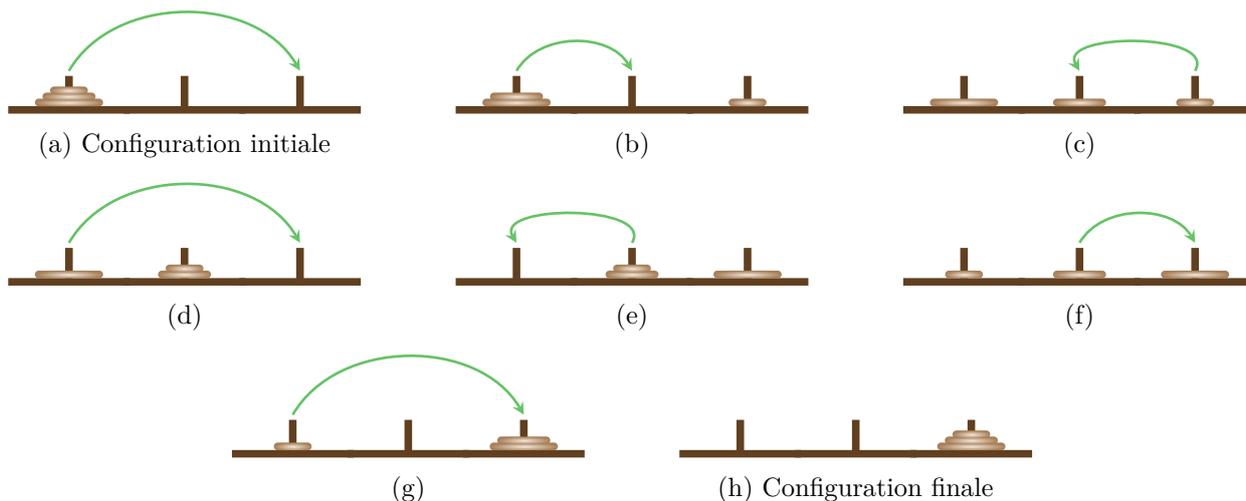


FIGURE 3 – Mouvements pour 3 palets.

Vous pouvez utiliser le script [http://www.normalesup.org/~dconduche/informatique/PC/Devoirs/DL1\\_2018\\_HanoiVisualisation.py](http://www.normalesup.org/~dconduche/informatique/PC/Devoirs/DL1_2018_HanoiVisualisation.py) pour visualiser une configuration de piquets, et donc une suite de mouvements.

Pour  $n$  palets, il suffit de savoir bouger  $n - 1$  palets vers le piquet central. Ensuite, on bouge le plus grand palet vers le piquet final, et de nouveau on sait bouger  $n - 1$  palets vers le piquet final :

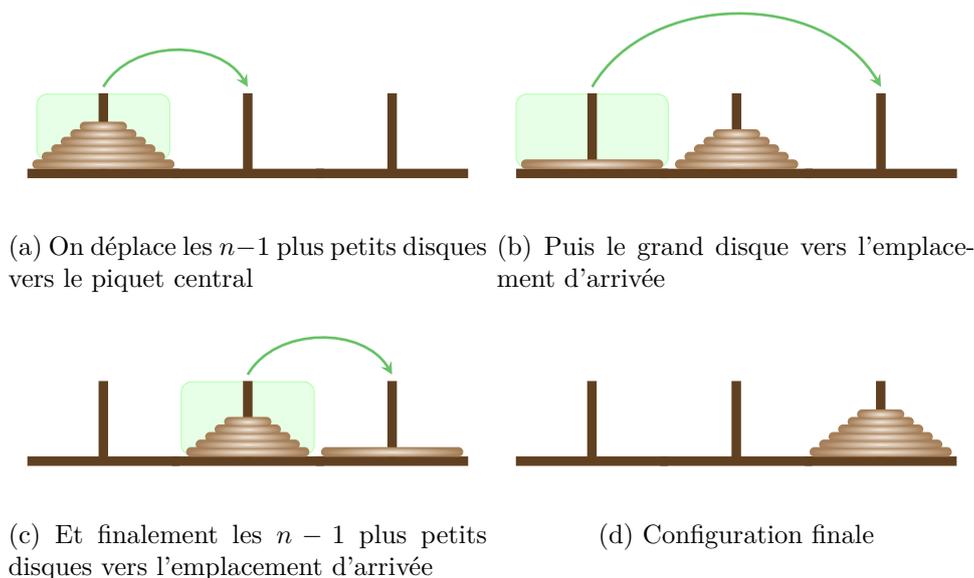


FIGURE 4 – Mouvements

- 1) On se donne un entier  $n$  (nombre de disques). Générer la liste  $L=[n, n-1, \dots, 1]$ , puis la liste **piquets** dans l'état initial, c'est-à-dire  $[L, [], []]$ .
- 2) Par la suite, les trois piquets seront numérotés  $p, q$  et  $r$  avec  $(p, q, r) \in \{0, 1, 2\}^3$  mais tous différents deux à deux. Écrire une fonction **deplace**( $p, q$ ) qui déplace le dernier disque du piquet  $p$  vers le piquet  $q$ , en l'insérant au sommet de la pile de disques.
- 3) On souhaite écrire une fonction récursive **hanoi**( $n, p, q, r$ ) qui déplace toute une pile de  $n$  disques initialement situés sur le piquet  $p$  vers le piquet  $r$ , en utilisant le piquet  $q$ .  
Par exemple avec l'état initial, l'appel de **Hanoi**( $3, 0, 1, 2$ ) conduit aux mouvements représentés à la figure 3, et **Hanoi**( $6, 0, 1, 2$ ) conduit à l'état final représenté à la figure 2. Écrire cette fonction de façon totalement récursive, la récursivité portant sur le nombre  $n$  de disques de la pile.
- 4) Combien de déplacements sont nécessaires pour que les  $n$  disques passent tous du piquet de « départ » au piquet d'« arrivée » ?
- 5) Selon la légende imaginée par E.Lucas, le jeu a commencé en - 1 500 avant JC, dans un temple à Hanoï, avec 64 disques d'or et la fin du monde arrivera quand le jeu se termine.  
Le temps nécessaire pour terminer le jeu est-il de l'ordre de l'âge de l'Univers (14 milliards d'années) ?  
On prendra une seconde par déplacement.

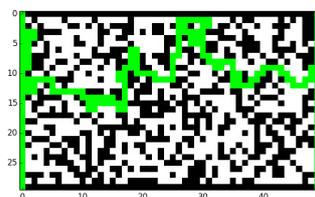
## Exercice 2 (Percolation)

En général, la percolation est un phénomène de transfert d'un liquide à travers un milieu poreux, qui y offre une certaine résistance. Nous allons discuter un modèle très simplifié, en deux dimensions.

Considérons deux « électrodes » (deux segments, en vert sur les dessins ci-après) éloignées, et séparées par un milieu, un rectangle composé de plusieurs carrés, « noirs » (isolants) ou « blancs » (conducteurs). Leur distribution est aléatoire, uniforme dans le rectangle, mais la densité des conducteurs peut être différente des isolants. La probabilité qu'un carré soit conducteur, par exemple 60%, est établie comme une constante globale par l'expérimentateur (dans la vraie percolation, le milieu offre une « résistance » variable au flux). La question est : est-ce que le courant passe entre les électrodes, c'est-à-dire peut-on trouver un chemin entre les deux électrodes qui ne passent que par des carrés conducteurs (si oui, on dira que le système est *globalement conducteur*, si non *globalement isolant*) ?



Les deux dessins ci-dessus ont tous les deux été fait avec chaque fois une probabilité de 60% qu'un carré donné soit conducteur. Le système de droite est globalement isolant, alors que celui de gauche est globalement conducteur (on suppose que le courant ne peut pas passer entre deux carrés conducteurs qui ne se touchent que par un coin ; il faut une arête en commun pour que le courant passe). Ci dessous un chemin reliant les deux électrodes pour le système de gauche :



On peut représenter notre milieu par un tableau numpy `systeme` de taille  $N \times M$  (où  $N$  est le nombre de lignes,  $M$  celui de colonnes,  $N = 30$ ,  $M = 50$  dans les dessins ci-dessus), en remplissant chaque case par la valeur d'une variable de Bernoulli de paramètre  $p$  ( $p = 0.6$  dans les dessins ci-dessus), en supposant chaque case indépendante des autres.

Pour simplifier (cela évitera des tests du débordement des indices), on supposera les bords verticaux conducteurs : `systeme[:,0]=systeme[:,-1]=0`. Puis les bords horizontaux isolants : `systeme[0,:]=systeme[-1,:]=1` (on ne tient pas compte pour le moment de la couleur verte).

- 1) Créer un tableau `systeme` qui respecte les conditions précédentes, et le faire afficher. *On rappelle que `np.random.rand(N, M)` crée un tableau de la forme donnée contenant des flottants aléatoires issus d'une distribution uniforme sur  $[0, 1[$ .*
- 2) Un chemin conducteur dans notre milieu est une suite de cases blanches contiguës, qui part de la gauche (coordonnées de la forme  $(i, 0)$ ) pour arriver à droite (coordonnées de la forme  $(i, -1)$ ). Créer une fonction

`teste(chemin, systeme)`

qui teste si une liste `chemin` de points de `systeme` est un chemin conducteur. Les éléments de `chemin` sont des couples de coordonnées  $(i, j)$  du tableau `systeme`.

- 3) Écrire une fonction `affChem(chemin, systeme)` qui affiche le chemin dans le tableau `systeme`.
- 4) Méthode naïve : Les chemins qui nous intéressent ne peuvent pas repasser par une case où il sont déjà passés : cela crée une boucle inutile. Une méthode naïve consiste donc à dénombrer toutes les suites *quelconques*<sup>1</sup> de cases de la grille, sans jamais prendre deux fois la même case, et à tester si ce sont des chemins conducteurs. On peut restreindre (un peu) le nombre de possibilité en ignorant les première et dernière colonnes (qui ne servent à rien), et en imposant à la première case et à la dernière case d'être dans les bonnes colonnes (respectivement 1 et  $M - 1$ ).
- a) Écrire une fonction `cheminNaif(systeme)` qui trouve un éventuel chemin conducteur par la méthode naïve. On prendra garde à fixer un  $N$  et un  $M$  vraiment petits.

Elle retournera un chemin au format décrit ci-dessus ou `False` s'il n'y en a pas.

Indication : On pourra utiliser la fonction `permutations` du package `itertools`, ou coder une fonction permutation récursive pour les plus motivés.

---

1. C'est une méthode naïve.

- b) Déterminer la complexité dans le pire cas de cet algorithme. Est-il utilisable dans l'exemple donné au début ?
- 5) Méthode par *backtracking* : Cherchons une méthode plus efficace pour trouver un chemin dans le conducteur. On part d'une case blanche de la colonne 1, vers la droite. En chaque case, il y a (potentiellement) jusqu'à 4 directions possibles. On en choisit une, et on poursuit son chemin. Si on se rend compte au bout d'un certain nombre d'étapes qu'il est mauvais, on a besoin d'une méthode pour revenir en arrière dans ce chemin, afin d'en emprunter un autre. On utilise une pile. Le chemin empile successivement les points par lesquels il passe. Si un point est la fin d'un mauvais chemin (une impasse : pas de voisins, ou les voisins ont déjà été essayés sans succès, et ont donc été marqués comme « mauvais »), on le dépile, on le marque comme « mauvais » (géré par une liste de booléens par exemple), et on essaye un autre chemin à partir du point d'avant. Si aucun bon chemin ne part du point d'avant, on le dépile encore, et on essaye d'autres chemins partant de son prédécesseur. etc ... C'est ce qu'on appelle le *backtracking*.
- a) Écrire une fonction `conducteur(systeme)` qui renvoie `True` si le système est globalement conducteur, `False` sinon, basée sur le principe du *backtracking*.
- b) Modifier la fonction précédente pour afficher un chemin entre les deux électrodes s'il y en a un.
- c) Pour aller plus loin, on revient à la version de la fonction écrite au a. On fait alors plusieurs simulations (quelques dizaines ou centaines) à  $p$  fixé (c'est-à-dire que l'on teste plusieurs systèmes), et on évalue la fréquence de réalisation de l'apparition de la conductance du système. On trace ensuite cette fonction de  $p$  (pour  $0,45 \leq p \leq 0,75$ ).

**Remarque.** On peut alors constater que le système n'augmente pas sa conductance de manière douce et régulière quand la densité  $p$  augmente. En dessous de 0,55 il ne se passe rien, le système reste (presque!) toujours isolant. D'autre part, au dessus de 0,67 il devient presque toujours conducteur ! On appelle cela le phénomène du seuil. Pour un système très grand, et le nombre d'échantillons également très grand, le seuil devient très dramatique, autour de la densité 0,6. (Selon une source, la vraie valeur pour un système infini est 0,59275) ...

Le phénomène du seuil possède des affinités avec les transitions de phase en physico-chimie, et on le voit dans d'autres circonstances. Supposons qu'un réseau de vaisseaux sanguins subit une lente obstruction par des couches de cholestérol qui s'accumulent, s'accumulent... On peut avoir l'impression que le flux du sang subira la résistance de plus en plus grande, et que ceci sera facile à diagnostiquer. Mais non, le flux passera, passera, et ... au dessus d'un certain seuil d'impureté, il y aura un blocage global. L'infarctus, l'accident cérébral, etc. Mais on voit les mêmes phénomènes dans la géologie, ou dans le processus de solidification de la gelée quand la température baisse. □