

# Tris, complexité et dichotomie

Bonus  
04

## Exercice 1

Le tri stupide est un algorithme de tri qui consiste à mélanger un tableau aléatoirement jusqu'à ce qu'il soit trié.

1. Écrire une fonction `tri_stupide(tab)` qui implémente le tri stupide (on pourra utiliser le mélange de Yates-Fisher).
2. En moyenne, combien de mélange seront-ils nécessaires pour trier le tableau ?
3. Un ordinateur effectue  $\sim 10^9$  calculs par seconde. En supposant qu'un seul (!) calcul permet de générer et vérifier un mélange, combien de temps sera nécessaire pour trier un tableau de longueur 20 ?

## Exercice 2

Un autre algorithme de mélange est le même que celui utilisé pour mélanger un jeu de cartes : on procède à des coupes successives.

1. Écrire une fonction `coupe(tab, i)` qui « coupe » le tableau selon l'indice  $i$ .

Un exemple : prenons le tableau `tab = [0,1,2,3,4,5]`. Alors `coupe(tab, 2)` renverra `[2,3,4,5,0,1]`.

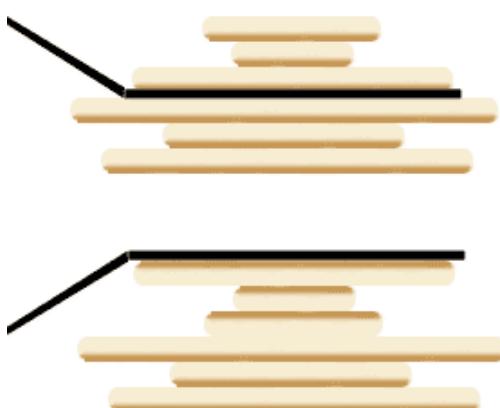
2. Écrire une fonction `melange_coupes(tab, k)` qui procède au mélange en utilisant  $k$  coupes aléatoires.
3. En partant du tableau `[0,1,2,3]`, déterminer l'ensemble de tous les tableaux accessibles par coupes successives : qu'en déduire ?

## Exercice 3

Une autre activité rentable pour mettre en pratique le tri des tableaux est la confection de crêpes.

Vous disposez de  $N$  crêpes, empilées les unes sur les autres. Vous souhaitez les trier (la plus large en bas, la plus petite en haut) (à noter : malgré les largeurs différentes, les petites crêpes peuvent parfaitement supporter les poids des grandes crêpes). Pour manipuler vos crêpes, vous disposez d'une spatule : vous pouvez la glisser sous n'importe quelle crêpe, récupérer le tas de crêpes au-dessus, retourner ce tas de crêpes et le reposer sur notre pile.

Écrire un algorithme permettant de trier la pile de crêpes. Dans le pire cas, combien de retournements aurez-vous besoin pour trier toutes vos crêpes ?



*Source : Wikipedia*