# Algorithmes probabilistes

François Schwarzentruber

8 mai 2023

**Définition 1** Un algorithme probabiliste est un algorithme qui peut, à plusieurs moments, prendre des décisions en fonction de tirages aléatoires uniformes $r \in \{1, \dots, R\}$ où les $R$ sont des entiers.

**Probablement rapide**
Algorithme de Las Vegas


Exemple : Tri rapide randomisé

**Probablement correct**
Algorithme de Monte Carlo

Exemples : Algorithme de Freivalds, algorithme de Karger

## 1 Tri rapide randomisé

☺ contrer le pire cas et avoir une bonne espérance de la complexité temporelle

```
fonction trirapide(T : ensemble)
    si |T| ≤ 1
        renvoyer T
    sinon
        e := choisir un pivot aléatoirement uniformément dans T
        renvoyer trirapide({b ∈ T | b < e}) :: e :: trirapide({b ∈ T | b > e})
```

**Théorème 2** Soit $X$ le nombre de comparaisons au cours de l'algorithme. On a : $\mathbb{E}(X) = O(n \log n)$.

## 2 Algorithme de Freivalds

Réf : [MR95], p. 162

☺ meilleure complexité
☹ mais avec une faible probabilité d'erreur

**Définition 3** **Vérification de la multiplication de matrices**

entrée : trois matrices carrés $A, B, C$ de taille $n \times n$ à valeurs dans un corps commutatif $\mathbb{K}$
sortie : oui si $AB = C$, non sinon.

**Proposition 4** Ce problème admet :

1. un algorithme naïf déterministe en temps $O(n^3)$ ;
2. l'algorithme déterministe diviser pour régner de Strassen, en temps $O(n^{2.807})$ ;
3. l'algorithme déterministe de Coppersmith–Winograd en temps $O(n^{2.376})$ [CW90].
4. un algorithme déterministe dû à Alman et Williams en $O(n^{2.372})$ [AW21]

Technique de l'empreinte : au lieu de vérifier $AB = C$, on vérifie $ABx = Cx$ pour un $x$ choisi aléatoirement

```
fonction freivalds(A, B, C)
    choisir un vecteur x ∈ {0,1}^n aléatoirement de manière uniforme
    si A(Bx) = Cx renvoyer oui sinon renvoyer non
```

**Proposition 5** L'algorithme probabiliste $freivalds$ est en temps $O(n^2)$.

**Proposition 6**
— Si $AB = C$ alors $freivalds(A, B, C)$ renvoie toujours oui.
— Si $AB \neq C$ alors $\mathbb{P}(freivalds(A, B, C) \text{ renvoie oui}) \leq \frac{1}{2}$.

# 3 Algorithme de Karger pour calculer une coupe minimale

Réf : [MR95][p. 7, 24, p. 289]

☺ Algorithme de Karger simple à comprendre, alors qu'il faut se lever de bonne heure pour comprendre :
flots, algorithme de Ford-Fulkerson, graphe résiduel, chemin améliorant, théorème de dualité coupe min/flot max, boucle pour toute destination, etc.

☺ Optimisations qui donnent l'algorithme le plus efficace connu à ce jour
☺ Algorithme qui, en le répétant, donne des solutions de plus en plus bonnes
☹ Pas de garantie d'avoir la meilleure solution

## 3.1 Définitions

**Définition 7 (coupe)** Soit $G = (V, E)$ un graphe, connexe, non orienté, avec arêtes multiples mais sans boucles. On appelle coupe de $G$ toute partition $\{X, Y\}$ de $G$ avec $X \neq \emptyset$ et $Y \neq \emptyset$.

**Définition 8 (arêtes d'une coupe)** L'ensemble des arêtes d'une coupe $\{X, Y\}$ est

$$C_{\{X,Y\}} = \{(x, y) \in E \mid x \in X \text{ et } y \in Y\}.$$

**Définition 9 (taille d'un coupe)** La taille d'une coupe $\{X, Y\}$ est le cardinal de $C_{\{X,Y\}}$.

**Définition 10 (coupe minimale)** Une coupe $\{X, Y\}$ est minimale si $C_{\{X,Y\}}$ est de cardinal minimal.

**Définition 11** **Problème du calcul de la coupe minimale**
entrée : Un graphe $G$ connexe, non orienté, avec arêtes multiples, sans boucles
sortie : une coupe minimale de $G$

**Proposition 12** Le problème du calcul de la coupe minimale est dans P.

## 3.2 Description de l'algorithme

```
fonction algoKarger(G)
    tant que G a strictement plus de 2 sommets
        sélectionner uniformément au hasard une arête e de G
        contracter l'arête e
    renvoyer les arêtes de G
```

**Exercice 13** Montrer que l'on peut implémenter l'algorithme de Karger en temps $O(n^2)$, où $n$ est le nombre de sommets.

## 3.3 Correction

**Proposition 14** Pour tout graphe $G$ de $n$ sommets, la probabilité que la coupe retournée soit minimale est $\geq \frac{2}{n^2}$.

Heureusement, en itérant on peut avoir avec une forte probabilité une coupe minimale. Plus précisément :

**Proposition 15** Pour tout $\epsilon \in ]0, 1[$, on a une probabilité $\geq 1 - \epsilon$ d'avoir calculé une coupe minimale avec au moins $\frac{n^2}{2} \ln \frac{1}{\epsilon}$ répétitions indépendantes de l'algorithme sur le graphe initial $G$.

**Exercice 16** Montrer qu'il y a au plus $\frac{n(n-1)}{2}$ coupes minimales.

# Remerciements

Merci à Arnaud Jobin pour les discussions sur la démonstration actuelle du temps d'exécution du tri rapide.

# Références

[AW21] Josh Alman and Virginia Vassilevska Williams. A refined laser method and faster matrix multiplication. In Dániel Marx, editor, *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms, SODA 2021, Virtual Conference, January 10 - 13, 2021*, pages 522–539. SIAM, 2021.

[CW90] Don Coppersmith and Shmuel Winograd. Matrix multiplication via arithmetic progressions. *J. Symb. Comput.*, 9(3) :251–280, 1990.

[MR95] R. Motwani and P. Raghavan. *Randomized algorithms*. Cambridge university press, 1995.