Algorithmes probabilistes

François Schwarzentruber

2 mars 2022

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

Probablement rapide
Algorithme de Las Vegas

Exemple : Tri rapide randomisé

Probablement correct
Algorithme de Monte Carlo

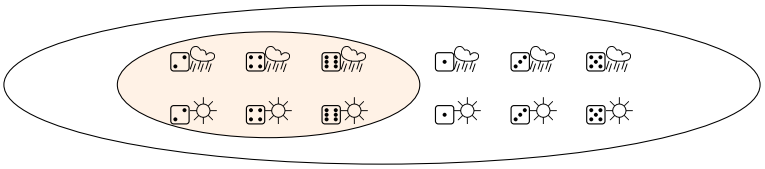
Exemples : Algorithme de Freivalds, algorithme de Karger

1 Rappels de probabilité

Soit Ω l'univers des mondes possibles.

Définition 2 (événement) Un événement est un sous-ensemble $A \subseteq \Omega$.

Exemple 3 $A :=$ 'Le dé affiche un nombre pair'.



Définition 4 (probabilité) La probabilité $\mathbb{P}(A)$ est la mesure de A, i.e. son aire.

Proposition 5 (probabilité d'une union disjointe) Soit A et B deux événements disjoints.

$$\mathbb{P}(A \sqcup B) = \mathbb{P}(A) + \mathbb{P}(B).$$

Définition 6 (probabilité conditionnelle) La probabilité conditionnelle de A sachant B est :

$$\mathbb{P}(A \mid B) := \frac{\mathbb{P}(A \cap B)}{\mathbb{P}(B)}.$$

Proposition 7 (probabilité de l'intersection d'événements) Pour $E_1, \ldots, E_n$ des événements quelconques :

$$\mathbb{P}(\bigcap_{j=1}^{n} E_j) = \mathbb{P}(E_1) \times \mathbb{P}(E_2 \mid E_1) \times \mathbb{P}(E_3 \mid E_1 \cap E_2) \times \cdots \times \mathbb{P}(E_n \mid \bigcap_{j=1}^{n-1} E_j).$$

Définition 8 (variable aléatoire) Une variable aléatoire à valeur dans $\mathbb{N}$ est une fonction $X : \Omega \to \mathbb{N}$.

Définition 9 (espérance) L'espérance d'une variable aléatoire X est sa moyenne :

$$\mathbb{E}(X) := \sum_{k \in \mathbb{N}} k \cdot \mathbb{P}(X{=}k).$$

Proposition 10 (linéarité de l'espérance) $\mathbb{E}(X + Y) = \mathbb{E}(X) + \mathbb{E}(Y)$.

Proposition 11 Soit X à valeur dans $\{0, 1\}$. Alors $\mathbb{E}(X) = \mathbb{P}(X{=}1)$.

2 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})
```

Exercice 1

1. Faire venir quelqu'un au tableau pour expliquer le tri rapide avec des magnets
2. Comment implémenter la notion d'ensemble ici ? (deux indices i, j dans le tableau)
3. Réfléchir à la notion d'algorithme en place sur l'exemple du tri rapide.
4. Écrire une version du tri rapide qui soit en place.

Théorème 12 Soit X le nombre de comparaisons au cours de l'algorithme. On a : $\mathbb{E}(X) = O(n \log n)$.

Démonstration.
Soit S la permutation triée de l'ensemble T, i.e $S_1 < S_2 < \cdots < S_n$. Pour tout $1 \leq i < j \leq n$, on pose :

$$X_{i,j} := \begin{cases} 1 & \text{si } S_i \text{ et } S_j \text{ sont comparés au cours de l'exécution du tri rapide randomisé} \\ 0 & \text{sinon.} \end{cases}$$

S_i et S_j sont comparés ssi l'un est choisi comme pivot. De plus, ils sont comparés au plus une fois car il n'y aura plus de comparaisons avec le pivot. Ainsi :

Fait 13 $X = \sum_{1 \leq i < j \leq n} X_{i,j}$.

Fait 14 Soit $i < j$. S_i et S_j sont comparés ssi S_i ou S_j est le premier élément parmi $S_{i..j}$ à être pivot.

Démonstration. $\boxed{\Rightarrow}$ Supposons que S_i et S_j sont comparés. L'un des deux est donc pivot, disons S_i. Mais alors aucun élément dans $S_{i+1,..,j}$ n'a encore été pivot : sinon il aurait séparé S_i et S_j et S_i et S_j n'auraient pas été comparés. Donc S_i est le 1er élément dans $S_{i..j}$ à être pivot.
$\boxed{\Leftarrow}$ Réciproquement, si, disons, S_i est le premier élément dans $S_{i..j}$ à être pivot, cela signifie que S_i et S_j n'ont pas encore été séparé à ce moment là. Ainsi, le pivot S_i et S_j qui sont comparés. ■

$$\dfrac{\dfrac{\dfrac{\text{Fait 13}}{X = \sum_{1 \leq i < j \leq n} X_{i,j}}}{\mathbb{E}(X) = \sum_{1 \leq i < j \leq n} \mathbb{E}(X_{i,j})} \qquad \dfrac{\dfrac{\mathbb{E}(X_{i,j}) = \mathbb{P}(S_i \text{ et } S_j \text{ sont comparés}) \qquad \text{Fait 14}}{\mathbb{E}(X_{i,j}) = \mathbb{P}(S_i \text{ ou } S_j \text{ premier pivot dans } S_{i..j})} \qquad \text{Fait 15}}{\mathbb{E}(X_{i,j}) = \frac{1}{j-i+1} + \frac{1}{j-i+1} = \frac{2}{j-i+1}}}{\dfrac{\mathbb{E}(X) = \sum_{1 \leq i < j \leq n} \frac{2}{j-i+1} \leq 2n \sum_{k=1}^{n} \frac{1}{k}}{\mathbb{E}(X) = O(n \log n)}}$$

PS : pour le calcul, faire un dessin avec les intervalles.

Fait 15 Soit a un élément de $S_{i..j}$. $\mathbb{P}(a \text{ premier pivot dans } S_{i..j}) = \frac{1}{j-i+1}$.

Démonstration. Posons $P(k)$ la propriété suivante, que l'on montre par récurrence sur k :

> Pour tout sous-tableau T' de taille $\leq k$ qui contient tous les éléments de $S_{i..j}$, la probabilité que a soit le premier pivot parmi $S_{i..j}$ dans l'un des sous-appels récursifs depuis $QS(T')$ est $\frac{1}{j-i+1}$.

Montrons $P(0)$. Un tableau vide ne contient pas les éléments de $S_{i..j}$, d'où $P(0)$.
Supposons $P(k-1)$ et montrons $P(k)$. Soit T' de taille k qui contient tous les éléments de $S_{i..j}$. Calculons la probabilité que a soit le premier pivot parmi $S_{i..j}$ dans l'un des sous-appels récursifs de $QS(T')$. Examinons quel pivot principal on tire dans l'appel $QS(T')$:

1. On tire a comme pivot (et donc comme premier pivot parmi $S_{i..j}$) avec une probabilité $\frac{1}{k}$;
2. On tire un pivot de $S_{i..j}$ différent de a. Mais alors a n'est pas premier pivot : il y a une contribution de 0.

3. On tire un pivot hors de $S_{i..j}$ avec une probabilité $\frac{k-(j-i+1)}{k}$. Ce pivot sépare le tableau en 2 sous-tableaux dont l'un des deux – notons le T'' – contient tous les éléments de $S_{i..j}$. Si a est choisi comme pivot c'est dans l'appel $QS(T'')$. La taille de T'' est $< k$. Par hypothèse de récurrence $P(k-1)$, la probabilité que a soit le premier pivot parmi $S_{i..j}$ dans les sous-appels à partir de $QS(T'')$ est $\frac{1}{j-i+1}$.

Ainsi la probabilité cherchée vaut $\frac{1}{k} + 0 + \frac{k-(j-i+1)}{k} \times \frac{1}{(j-i+1)} = \frac{1}{(j-i+1)}$.
Plus formellement il faut calculer $\mathbb{P}(a$ premier pivot dans $S_{i..j}$ durant $QS(T'))$. C'est la somme de :

1. $\mathbb{P}(a$ premier pivot dans $S_{i..j}$ durant $QS(T')$ et a est le pivot choisi directement dans $QS(T'))) = 1/k$.
2. $\mathbb{P}(a$ premier pivot dans $S_{i..j}$ durant $QS(T')$ et élément $\neq a$ de $S_{i..j}$ est le pivot choisi directement dans $QS(T'))) = 0$.
3. $\mathbb{P}(a$ premier pivot dans $S_{i..j}$ durant $QS(T')$ et un autre élément hors de $S_{i..j}$ comme pivot choisi directement dans Q
qui vaut $\mathbb{P}(a$ premier pivot dans $S_{i..j}$ durant $QS(T'')$ où $|T''| \leq k-1$ et élément hors de $S_{i..j}$ comme pivot choisi dire
$= \mathbb{P}(a$ premier pivot dans $S_{i..j}$ durant $QS(T'')$ où $|T''| \leq k-1) \times \mathbb{P}($élément hors de $S_{i..j}$ comme pivot choisi directem
$= \frac{1}{j-i+1} - - - - \times \frac{k-(j-i+1)}{k}$

■ ■

3 Algorithme de Freivalds

Réf : [MR95], p. 162

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

Soit $\mathbb{K}$ un corps commutatif.

Définition 16 **Vérification de la multiplication de matrices**
entrée : trois matrices carrés A, B, C de taille $n \times n$ à valeurs dans $\mathbb{K}$
sortie : oui si $AB = C$, non sinon.

Proposition 17 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]

Proposition 18 L'algorithme probabiliste suivant, $freivalds$, est en temps $O(n^2)$.

Démonstration.
— Le choix du vecteur aléatoire x est en $O(n)$.
— Suivent trois produits matrices-vecteurs qui sont en $O(n^2)$.
■

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 19
— Si $AB = C$ alors $freivalds(A, B, C)$ renvoie toujours oui.
— Si $AB \neq C$ alors $\mathbb{P}(freivalds(A, B, C)$ renvoie oui$) \leq \frac{1}{2}$.

Démonstration. Posons $D := C - AB$. La matrice D est non nulle donc il existe $i, j \in \{1, \ldots, n\}$ tel que $d_{ij} \neq 0$. Quitte à changer les indices, on peut supposer que $d_{11} \neq 0$. Pour toute valeur de $x_2, \ldots, x_n$, il n'y a qu'une valeur pour x_1 qui rende $(Dx)_1 = \sum_{j=1}^{n} x_j d_{1j}$ (i.e. la première coordonnée de Dx) nulle : il s'agit de $-\frac{1}{d_{11}} \sum_{j=2}^{n} x_j d_{1j}$. Ainsi étant donné, des valeurs $v_2, \ldots, v_n \in \{0, 1\}$ pour $x_2, \ldots, x_n$, on a :

$$\mathbb{P}(\sum_{j=1}^{n} x_j d_{1j} = 0 \mid x_{2..n} = v_{2..n}) \leq \frac{1}{2}$$

Mais alors :

$$\begin{aligned}
\mathbb{P}(freivalds(A,B,C) \text{ renvoie oui}) &= \mathbb{P}(Dx = 0) \\
&\leq \mathbb{P}((Dx)_1 = 0) \\
&\leq \mathbb{P}(\sum_{j=1}^{n} x_j d_{1j} = 0) \\
&= \sum_{v_{2..n} \in \{0,1\}^{n-1}} \mathbb{P}(\sum_{j=1}^{n} x_j d_{1j} = 0 \mid x_{2..n} = v_{2..n}) \times \mathbb{P}(x_{2..n} = v_{2..n}) \\
&\leq \frac{1}{2}.
\end{aligned}$$

■

Proposition 20 Si $AB \neq C$, la probabilité de ne pas détecter que $AB \neq C$ avec k exécutions indépendantes de $freivalds$ est $\leq \frac{1}{2^k}$.

Proposition 21 Il faut lancer $\log(1/\epsilon)$ fois l'algorithme pour avoir une probabilité d'erreur $\leq \epsilon$.

Démonstration. On cherche k minimum tel que $\frac{1}{2^k} \leq \epsilon$, $2^k \geq \epsilon$. Il s'agit de $k = \lceil \log \frac{1}{\epsilon} \rceil$. ■

4 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

4.1 Définitions

Définition 22 (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 23 (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 24 (taille d'un coupe) La taille d'une coupe $\{X, Y\}$ est le cardinal de $C_{\{X,Y\}}$.

Définition 25 (coupe minimale) Une coupe $\{X, Y\}$ est minimale si $C_{\{X,Y\}}$ est de cardinal minimal.

Définition 26 **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 27 Le problème du calcul de la coupe minimale est dans P.

4.2 Applications

On trouve quelques applications dans [KS96] que je résume ici.

Application 28 (réseaux de communication) Nombre minimal d'arêtes à couper pour que deux ordinateurs ne puissent plus communiquer.

Application 29 (documents avec liens hypertextes) La coupe minimale sépare les documents en deux catégories qui sont peu reliées, et donc qui, a priori sont à propos de sujets différents.

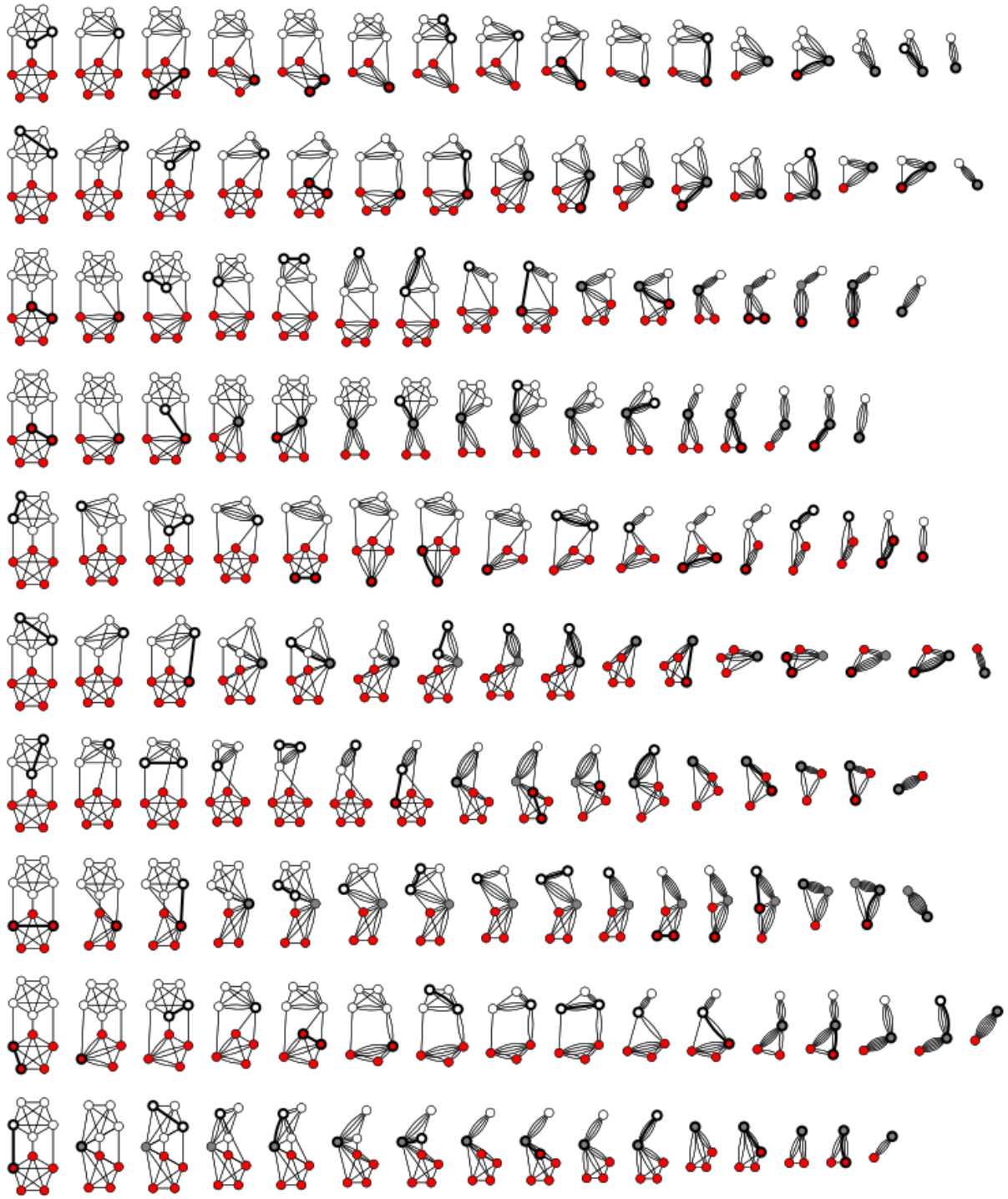
Application 30 (génie logiciel) Une coupe minimale dans le graphe de dépendances entre classes permet de découper le logiciel en deux packages.

Application 31 (compilation de langages parallèles) Considérons le graphe du programme où les nœuds sont les actions du programme et les arcs sont les communications (échange de message) entre points du programme. Il s'agit de minimiser le nombre d'échanges de message.

4.3 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
```

Durant l'algorithme, un sommet représente un sous-ensemble de sommets du graphe initial.



Source : wikipedia

Exercice 32 Montrer que l'on peut implémenter l'algorithme de Karger en temps $O(n^2)$, où n est le nombre de sommets.

4.4 Correction

Proposition 33 Pour tout graphe G de n sommets, la probabilité que la coupe retournée soit minimale est $\geq \frac{2}{n^2}$.

Démonstration. Soit C une coupe minimale de G et soit k sa taille.

Événements

E := algo renvoie C = algo ne choisit jamais une arête de C
E_i := algo ne choisit pas d'arêtes de C à la i-ème étape

On va montrer que $\mathbb{P}(E) \geq \frac{2}{n^2}$. Comme $E = \bigcap_{i=1}^{n-2} E_i$, on a :

$$\mathbb{P}(E) = \mathbb{P}(E_1) \times \mathbb{P}(E_2 \mid E_1) \times \mathbb{P}(E_3 \mid E_1 \cap E_2) \times \cdots \times \mathbb{P}(E_{n-2} \mid \bigcap_{j=1}^{n-3} E_j)$$

On a :

$$\frac{\mathbb{P}(E_1) = 1 - \frac{k}{|E|} \qquad \frac{\sum_{w \in V} deg(w) = 2|E| \qquad deg(w) \geq k}{G \text{ a au moins } \frac{kn}{2} \text{ arêtes}}}{\mathbb{P}(E_1) \geq 1 - \frac{2}{n}}$$

De manière générale, si $\bigcap_{j=1}^{i-1} E_j$ s'est produit, dans le graphe contracté de $n - i + 1$ sommets, une coupe minimale de ce nouveau graphe est toujours de taille k. Ainsi, on a :

$$\mathbb{P}(E_i \mid \bigcap_{j=1}^{i-1} E_j) \geq 1 - \frac{2}{n-i+1}$$

Finalement :

$$\mathbb{P}(E) \geq \Pi_{i=1}^{n-2} \left(1 - \frac{2}{n-i+1}\right) = \Pi_{i=1}^{n-2} \left(\frac{n-i-1}{n-i+1}\right) = \frac{(n-2) \times \cdots \times 2 \times 1}{n(n-1) \times \cdots \times 3} = \frac{2}{n(n-1)} \geq \frac{2}{n^2}.$$

■

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

Proposition 34 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.

Démonstration. La probabilité qu'une exécution ne donne pas une solution optimale est $\leq 1 - \frac{2}{n^2}$.
La probabilité qu'aucune exécution parmi k exécutions indépendantes ne donne une solution optimale est $\leq \left(1 - \frac{2}{n^2}\right)^k$.
Or pour tout $x > 0$,

$$\left(1 - \frac{1}{x}\right)^x < \frac{1}{e}.$$

Ainsi avec $x = \frac{n^2}{2}$.

$$\left(1 - \frac{2}{n^2}\right)^{\frac{n^2}{2}} < \frac{1}{e}$$

Donc en mettant à la puissance $\ln \frac{1}{\epsilon}$,

$$\left(1 - \frac{2}{n^2}\right)^{\frac{n^2}{2} \ln\left(\frac{1}{\epsilon}\right)} < e^{-\ln\left(\frac{1}{\epsilon}\right)} = \epsilon.$$

■

Exercice 35 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.

[KS96] David R Karger and Clifford Stein. A new approach to the minimum cut problem. *Journal of the ACM (JACM)*, 43(4) :601–640, 1996.

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