Algorithmique, structures informatiques et cryptologie

Loïck Lhote
loick.lhote@unicaen.fr

UNIVERSITÉ
CAEN
NORMANDIE

UNIVERSITÉ
CAEN
NORMANDIE

2- Introduction à l'algorithmique probabiliste

UNIVERSITÉ
CAEN
NORMANDIE

2- Introduction à l'algorithmique probabiliste

Intérêts des algorithmes probabilistes

Intérêts des algorithmes probabilistes

UNIVERSITÉ CAEN NORMANDIE

- Multiples algorithmes probabilistes au quotidien
  - Jeux de hasard ou de stratégie sur ordinateur
  - Tests de programmes
  - Génération de clés de sécurité pour les communications
  - ...
- On ne peut être certain du résultat d'un algorithme déterministe

From Wikipedia:

> Studies by IBM in the 1990s suggest that computers typically experience about one cosmic-ray-induced error per 256 megabytes of RAM per month.[15]

This means a probability of 3.7×10^{-9} per byte per month, or 1.4×10^{-15} per byte per second. If your program runs for 1 minute and occupies 20 MB of RAM, then the failure probability would be

$$1 - (1 - 1.4e\text{-}15)^{60 \times 20 \times 1024^2} = 1.8e\text{-}6 \text{ a.k.a. "5 nines"}$$

Error checking can help to reduce the aftermath of failure. Also, because of more compact size of chips as commented by Joe, the failure rate could be different from what it was 20 years ago.

Source : https://stackoverflow.com/questions/2580933/cosmic-rays-what-is-the-probability-they-will-affect-a-program

Intérêts des algorithmes probabilistes

UNIVERSITÉ CAEN NORMANDIE

Exercice : pour tester la primalité d'un nombre, nous disposons

- d'un algorithme probabiliste de complexité $O(n^3)$ pour des entiers de n bits mais qui se trompe avec une probabilité de $1/2^{100}$
- d'un algorithme déterministe de complexité $O(n^6)$ pour des entiers de n bits
- les deux algorithmes utilisent $1Ko$ de mémoire
- d'un ordinateur pouvant effectuer 10^{10} opérations par seconde (4 cœurs à 2.5GHz)

Question : Quel algorithme conseillez-vous pour $n = 2048$?

Rappel : probabilité d'une erreur cosmique $= 1.47 \times 10^{-12}$ erreur/Ko/s

Intérêts des algorithmes probabilistes

UNIVERSITÉ CAEN NORMANDIE

Exercice : pour tester la primalité d'un nombre, nous disposons

- d'un algorithme probabiliste de complexité $O(n^3)$ pour des entiers de n bits mais qui se trompe avec une probabilité de $1/2^{100}$
- d'un algorithme déterministe de complexité $O(n^6)$ pour des entiers de n bits
- les deux algorithmes utilisent $1Ko$ de mémoire
- d'un ordinateur pouvant effectuer 10^{10} opérations par seconde (4 cœurs à 2.5GHz)

Question : Quel algorithme conseillez-vous pour $n = 2048$?

Rappel : probabilité d'une erreur cosmique $= 1.47 \times 10^{-12}$ erreur/Ko/s

alg. det. : $\mathbb{P}(\text{erreur}) = 1 - (1 - 1.47 \times 10^{-12})^{2048^6/10^{10}} \approx 0.01$ (calcul : 360 ans ! ! !)

Intérêts des algorithmes probabilistes

UNIVERSITÉ CAEN NORMANDIE

Exercice : pour tester la primalité d'un nombre, nous disposons

- d'un algorithme probabiliste de complexité $O(n^3)$ pour des entiers de n bits mais qui se trompe avec une probabilité de $1/2^{100}$
- d'un algorithme déterministe de complexité $O(n^6)$ pour des entiers de n bits
- les deux algorithmes utilisent $1Ko$ de mémoire
- d'un ordinateur pouvant effectuer 10^{10} opérations par seconde (4 cœurs à 2.5GHz)

Question : Quel algorithme conseillez-vous pour $n = 2048$?

Rappel : probabilité d'une erreur cosmique $= 1.47 \times 10^{-12}$ erreur/Ko/s

alg. det. : $\mathbb{P}(\text{erreur}) = 1 - (1 - 1.47 \times 10^{-12})^{2048^6/10^{10}} \approx 0.01$ (calcul : 360 ans ! ! !)

alg. prob. : $\mathbb{P}(\text{erreur}) \approx 1 - (1 - 1.47 \times 10^{-12})^{2048^3/10^{10}} + 2^{-100} \approx 1.3 \times 10^{-12}$

Intérêts des algorithmes probabilistes

- Quitte à faire des concessions
  - sur la certitude du temps de calcul et/ou
  - sur la qualité du résultat

  les algorithmes probabilistes sont
  - souvent très efficaces en temps
  - et souvent très précis dans les résultats

Intérêts des algorithmes probabilistes

UNIVERSITÉ CAEN NORMANDIE

- Quitte à faire des concessions
  - sur la certitude du temps de calcul et/ou
  - sur la qualité du résultat

  les algorithmes probabilistes sont
  - souvent très efficaces en temps
  - et souvent très précis dans les résultats

Difficulté : concevoir et analyser ces algorithmes

Intérêts des algorithmes probabilistes

UNIVERSITÉ CAEN NORMANDIE

- Quitte à faire des concessions
  - ☐ sur la certitude du temps de calcul et/ou
  - ☐ sur la qualité du résultat

  les algorithmes probabilistes sont
  - ☐ souvent très efficaces en temps
  - ☐ et souvent très précis dans les résultats

Difficulté : concevoir et analyser ces algorithmes

- Notez que la notion de sécurité en cryptologie est naturellement basée sur les algorithmes probabilistes :

  "avoir une chance non négligeable de cracker un protocole"

Intérêts des algorithmes probabilistes

UNIVERSITÉ CAEN NORMANDIE

- Quitte à faire des concessions
  - ☐ sur la certitude du temps de calcul et/ou
  - ☐ sur la qualité du résultat

  les algorithmes probabilistes sont
  - ☐ souvent très efficaces en temps
  - ☐ et souvent très précis dans les résultats

Difficulté : concevoir et analyser ces algorithmes

- Notez que la notion de sécurité en cryptologie est naturellement basée sur les algorithmes probabilistes :

  "avoir une chance non négligeable de cracker un protocole"

- L'objectif est de présenter plusieurs familles d'algorithmes probabilistes

Paradoxe des anniversaires

UNIVERSITÉ CAEN NORMANDIE

Amusons nous un peu...

Question : Combien d'élèves faut-il dans une classe pour avoir plus d'une chance sur 2 que deux élèves soient nés le même jour ?

Réponse :

Paradoxe des anniversaires

UNIVERSITÉ CAEN NORMANDIE

Amusons nous un peu...

Question : Combien d'élèves faut-il dans une classe pour avoir plus d'une chance sur 2 que deux élèves soient nés le même jour ?

Réponse : il faudra attendre la partie *Algorithmes Las-Vegas*...

Pari : Qui veut parier avec moi ?

UNIVERSITÉ
CAEN
NORMANDIE

2- Introduction à l'algorithmique probabiliste

Notions basiques de probabilité

Probabilités

UNIVERSITÉ CAEN NORMANDIE

On supposera connues les notions suivantes :

- distribution de probabilité,
- distribution uniforme,
- probabilités conditionnelles,
- variables aléatoires,
- espérance,
- variance ou écart-type,
- moments
- ...

Loi de Bernoulli

UNIVERSITÉ CAEN NORMANDIE

Loi de Bernoulli

Une variable aléatoire X suit une loi de Bernoulli de paramètre $p \in [0, 1]$ si

$$\mathbb{P}[X = 1] = p, \qquad \mathbb{P}[X = 0] = 1 - p.$$

L'espérance et la variance de X vérifient

$$\mathbb{E}[X] = p, \qquad \mathbb{V}(X) = p(1 - p).$$

Exemples :

Loi de Bernoulli

UNIVERSITÉ CAEN NORMANDIE

Loi de Bernoulli

Une variable aléatoire X suit une loi de Bernoulli de paramètre $p \in [0, 1]$ si

$$\mathbb{P}[X = 1] = p, \qquad \mathbb{P}[X = 0] = 1 - p.$$

L'espérance et la variance de X vérifient

$$\mathbb{E}[X] = p, \qquad \mathbb{V}(X) = p(1 - p).$$

Exemples :

- lancé d'une pièce (biaisée ou non) : *pile* $= 1$ et *face* $= 0$

$$[X = 1] = \text{le lancé est tombé sur } \textit{pile}$$

Loi de Bernoulli

UNIVERSITÉ CAEN NORMANDIE

Loi de Bernoulli

Une variable aléatoire X suit une loi de Bernoulli de paramètre $p \in [0, 1]$ si

$$\mathbb{P}[X = 1] = p, \qquad \mathbb{P}[X = 0] = 1 - p.$$

L'espérance et la variance de X vérifient

$$\mathbb{E}[X] = p, \qquad \mathbb{V}(X) = p(1 - p).$$

Exemples :

- lancé d'une pièce (biaisée ou non) : *pile* $= 1$ et *face* $= 0$

$$[X = 1] = \text{le lancé est tombé sur } \textit{pile}$$

- Un algorithme probabiliste donne la bonne (mauvaise) réponse avec une probabilité p quelle que soit l'entrée

Loi de Bernoulli

UNIVERSITÉ CAEN NORMANDIE

Considérons une pièce biaisée dont la probabilité de faire pile est 1/3.

- Quelle est la probabilité de faire face ?

Loi de Bernoulli

UNIVERSITÉ CAEN NORMANDIE

Considérons une pièce biaisée dont la probabilité de faire pile est $1/3$.

- Quelle est la probabilité de faire face ? $\mathbb{P}[\mathit{face}] = 1 - \frac{1}{3} = \frac{2}{3}$

Loi de Bernoulli

UNIVERSITÉ CAEN NORMANDIE

Considérons une pièce biaisée dont la probabilité de faire pile est $1/3$.

- Quelle est la probabilité de faire face ? $\mathbb{P}[face] = 1 - \frac{1}{3} = \frac{2}{3}$
- Quelle est la probabilité de faire successivement pile puis face ?

Loi de Bernoulli

Considérons une pièce biaisée dont la probabilité de faire pile est $1/3$.

- Quelle est la probabilité de faire face ? $\mathbb{P}[face] = 1 - \frac{1}{3} = \frac{2}{3}$
- Quelle est la probabilité de faire successivement pile puis face ?

$$\mathbb{P}[pile - face] = \mathbb{P}[pile]\mathbb{P}[face] = \frac{1}{3} \times \frac{2}{3} \qquad \text{(lancés indépendants)}$$

Loi de Bernoulli

Considérons une pièce biaisée dont la probabilité de faire pile est $1/3$.

- Quelle est la probabilité de faire face ? $\mathbb{P}[face] = 1 - \frac{1}{3} = \frac{2}{3}$
- Quelle est la probabilité de faire successivement pile puis face ?

$$\mathbb{P}[pile - face] = \mathbb{P}[pile]\mathbb{P}[face] = \frac{1}{3} \times \frac{2}{3} \qquad \text{(lancés indépendants)}$$

- Même question avec pile-pile-face-pile et face-pile-pile-pile ?

Loi de Bernoulli

UNIVERSITÉ CAEN NORMANDIE

Considérons une pièce biaisée dont la probabilité de faire pile est $1/3$.

- Quelle est la probabilité de faire face ? $\mathbb{P}[face] = 1 - \frac{1}{3} = \frac{2}{3}$
- Quelle est la probabilité de faire successivement pile puis face ?

$$\mathbb{P}[pile-face] = \mathbb{P}[pile]\mathbb{P}[face] = \frac{1}{3} \times \frac{2}{3} \qquad \text{(lancés indépendants)}$$

- Même question avec pile-pile-face-pile et face-pile-pile-pile ?

$$\mathbb{P}[p-p-f-p] = \mathbb{P}[f-p-p-p] = \frac{1}{3^3} \times \frac{2}{3} \qquad \text{(lancés indépendants)}$$

Loi de Bernoulli

UNIVERSITÉ CAEN NORMANDIE

Considérons une pièce biaisée dont la probabilité de faire pile est $1/3$.

- Quelle est la probabilité de faire face ? $\mathbb{P}[face] = 1 - \frac{1}{3} = \frac{2}{3}$
- Quelle est la probabilité de faire successivement pile puis face ?

$$\mathbb{P}[pile-face] = \mathbb{P}[pile]\mathbb{P}[face] = \frac{1}{3} \times \frac{2}{3} \qquad \text{(lancés indépendants)}$$

- Même question avec pile-pile-face-pile et face-pile-pile-pile ?

$$\mathbb{P}[p-p-f-p] = \mathbb{P}[f-p-p-p] = \frac{1}{3^3} \times \frac{2}{3} \qquad \text{(lancés indépendants)}$$

- En 4 lancés, quelle est la probabilité de faire 0,1,2,3,4 pile(s) ?

Loi de Bernoulli

Considérons une pièce biaisée dont la probabilité de faire pile est $1/3$.

- Quelle est la probabilité de faire face ? $\mathbb{P}[face] = 1 - \frac{1}{3} = \frac{2}{3}$
- Quelle est la probabilité de faire successivement pile puis face ?

$$\mathbb{P}[pile-face] = \mathbb{P}[pile]\mathbb{P}[face] = \frac{1}{3} \times \frac{2}{3} \qquad \text{(lancés indépendants)}$$

- Même question avec pile-pile-face-pile et face-pile-pile-pile ?

$$\mathbb{P}[p-p-f-p] = \mathbb{P}[f-p-p-p] = \frac{1}{3^3} \times \frac{2}{3} \qquad \text{(lancés indépendants)}$$

- En 4 lancés, quelle est la probabilité de faire 0,1,2,3,4 pile(s) ?
  Loi binomiale de paramètres $(4, 1/3)$:

$$\mathbb{P}[k \text{ piles}] = \frac{4!}{k!(4-k)!} \left(\frac{1}{3}\right)^k \left(\frac{2}{3}\right)^{n-k}$$

Loi binomiale

UNIVERSITÉ CAEN NORMANDIE

Loi binomiale

Une variable aléatoire X suit une loi de binomiale de paramètres (n, p) avec n entiers et $p \in [0, 1]$ si

$$\forall k \in \{1, 2, \dots, n\}, \qquad \mathbb{P}[X = k] = \binom{n}{k} p^k (1-p)^{n-k}.$$

L'espérance et la variance de X vérifient

$$\mathbb{E}[X] = np, \qquad \mathbb{V}(X) = np(1-p).$$

Exemples :

Loi binomiale

UNIVERSITÉ CAEN NORMANDIE

Loi binomiale

Une variable aléatoire X suit une loi de binomiale de paramètres (n, p) avec n entiers et $p \in [0, 1]$ si

$$\forall k \in \{1, 2, \dots, n\}, \qquad \mathbb{P}[X = k] = \binom{n}{k} p^k (1-p)^{n-k}.$$

L'espérance et la variance de X vérifient

$$\mathbb{E}[X] = np, \qquad \mathbb{V}(X) = np(1-p).$$

Exemples :

- Si $X_1, X_2, \dots, X_n$ sont n variables aléatoires i.i.d. de même loi de Bernoulli de paramètre p alors

$$X_1 + X_2 + \cdots + X_n \sim Binom(n, p).$$

Loi binomiale

UNIVERSITÉ CAEN NORMANDIE

Loi binomiale

Une variable aléatoire X suit une loi de binomiale de paramètres (n, p) avec n entiers et $p \in [0, 1]$ si

$$\forall k \in \{1, 2, \dots, n\}, \qquad \mathbb{P}[X = k] = \binom{n}{k} p^k (1-p)^{n-k}.$$

L'espérance et la variance de X vérifient

$$\mathbb{E}[X] = np, \qquad \mathbb{V}(X) = np(1-p).$$

Exemples :

- Si $X_1, X_2, \dots, X_n$ sont n variables aléatoires i.i.d. de même loi de Bernoulli de paramètre p alors

$$X_1 + X_2 + \cdots + X_n \sim Binom(n, p).$$

- lancé d'une pièce biaisée

Loi binomiale

UNIVERSITÉ CAEN NORMANDIE

- Considérons un problème de décision : la réponse est oui/non
- Considérons un algorithme qui pour toute entrée, retourne la bonne solution avec une probabilité $p = 2/3$
  (On ne sait pas ce qu'il retourne avec probabilité $1 - p = 1/3$)

Q1 : exécutons l'algorithme 1 fois sur une entrée donnée.
Donner une minoration de la probabilité de succès de l'algorithme.

Loi binomiale

- Considérons un problème de décision : la réponse est oui/non
- Considérons un algorithme qui pour toute entrée, retourne la bonne solution avec une probabilité $p = 2/3$
  (On ne sait pas ce qu'il retourne avec probabilité $1 - p = 1/3$)

Q1 : exécutons l'algorithme 1 fois sur une entrée donnée.
Donner une minoration de la probabilité de succès de l'algorithme.

Réponse : on pose X la variable aléatoire qui vaut 1 si l'algorithme répond correctement et 0 sinon.

Loi binomiale

- Considérons un problème de décision : la réponse est oui/non
- Considérons un algorithme qui pour toute entrée, retourne la bonne solution avec une probabilité $p = 2/3$
  (On ne sait pas ce qu'il retourne avec probabilité $1 - p = 1/3$)

Q1 : exécutons l'algorithme 1 fois sur une entrée donnée.
Donner une minoration de la probabilité de succès de l'algorithme.

Réponse : on pose X la variable aléatoire qui vaut 1 si l'algorithme répond correctement et 0 sinon.
Alors X suit une loi de Bernoulli de paramètre $p = 2/3$ et

Loi binomiale

- Considérons un problème de décision : la réponse est oui/non
- Considérons un algorithme qui pour toute entrée, retourne la bonne solution avec une probabilité $p = 2/3$
  (On ne sait pas ce qu'il retourne avec probabilité $1 - p = 1/3$)

Q1 : exécutons l'algorithme 1 fois sur une entrée donnée.
Donner une minoration de la probabilité de succès de l'algorithme.

Réponse : on pose X la variable aléatoire qui vaut 1 si l'algorithme répond correctement et 0 sinon.
Alors X suit une loi de Bernoulli de paramètre $p = 2/3$ et

$$\mathbb{P}[succes] = \mathbb{P}[X = 1] = p = 2/3 \quad (\text{en fait } p \geq 2/3)$$

Loi binomiale

- Considérons un problème de décision : la réponse est oui/non
- Considérons un algorithme qui pour toute entrée, retourne la bonne solution avec une probabilité $p = 2/3$.
  (On ne sait pas ce qu'il retourne avec probabilité $1 - p = 1/3$)

Q2 : Exécutons l'algorithme 3 fois sur une entrée donnée et retournons la réponse majoritaire.
Donner une minoration de la probabilité de succès de l'algorithme.

Loi binomiale

- Considérons un problème de décision : la réponse est oui/non
- Considérons un algorithme qui pour toute entrée, retourne la bonne solution avec une probabilité $p = 2/3$.
  (On ne sait pas ce qu'il retourne avec probabilité $1 - p = 1/3$)

Q2 : Exécutons l'algorithme 3 fois sur une entrée donnée et retournons la réponse majoritaire.
Donner une minoration de la probabilité de succès de l'algorithme.

Réponse : on pose X_i $(i = 1, 2, 3)$ la variable aléatoire qui vaut 1 si l'algorithme répond correctement à la i-ème exécution et 0 sinon.

Loi binomiale

- Considérons un problème de décision : la réponse est oui/non
- Considérons un algorithme qui pour toute entrée, retourne la bonne solution avec une probabilité $p = 2/3$.
  (On ne sait pas ce qu'il retourne avec probabilité $1 - p = 1/3$)

Q2 : Exécutons l'algorithme 3 fois sur une entrée donnée et retournons la réponse majoritaire.
Donner une minoration de la probabilité de succès de l'algorithme.

Réponse : on pose X_i $(i = 1, 2, 3)$ la variable aléatoire qui vaut 1 si l'algorithme répond correctement à la i-ème exécution et 0 sinon.

On pose $Y = X_1 + X_2 + X_3$.

Loi binomiale

- Considérons un problème de décision : la réponse est oui/non
- Considérons un algorithme qui pour toute entrée, retourne la bonne solution avec une probabilité $p = 2/3$.
  (On ne sait pas ce qu'il retourne avec probabilité $1 - p = 1/3$)

Q2 : Exécutons l'algorithme 3 fois sur une entrée donnée et retournons la réponse majoritaire.
Donner une minoration de la probabilité de succès de l'algorithme.

Réponse : on pose X_i ($i = 1, 2, 3$) la variable aléatoire qui vaut 1 si l'algorithme répond correctement à la i-ème exécution et 0 sinon.

On pose $Y = X_1 + X_2 + X_3$.

Alors Y suit une loi binomiale de paramètres $(3, p)$ et

Loi binomiale

- Considérons un problème de décision : la réponse est oui/non
- Considérons un algorithme qui pour toute entrée, retourne la bonne solution avec une probabilité $p = 2/3$.
  (On ne sait pas ce qu'il retourne avec probabilité $1 - p = 1/3$)

Q2 : Exécutons l'algorithme 3 fois sur une entrée donnée et retournons la réponse majoritaire.
Donner une minoration de la probabilité de succès de l'algorithme.

Réponse : on pose X_i $(i = 1, 2, 3)$ la variable aléatoire qui vaut 1 si l'algorithme répond correctement à la i-ème exécution et 0 sinon.

On pose $Y = X_1 + X_2 + X_3$.

Alors Y suit une loi binomiale de paramètres $(3, p)$ et

$$\mathbb{P}[succes] = \mathbb{P}[Y \geq 2] = \binom{3}{2} p^2(1-p) + \binom{3}{3} p^3 = \frac{20}{27} \approx 0.74$$

Loi binomiale

- Considérons un problème de décision : la réponse est oui/non
- Considérons un algorithme qui pour toute entrée, retourne la bonne solution avec une probabilité $p = 2/3$.
  (On ne sait pas ce qu'il retourne avec probabilité $1 - p = 1/3$)

Q3 : Refaites le même exercice avec $p = 1/2$ et $p = 1/3$. Que constatez-vous ?

Loi géométrique

- Considérons une pièce biaisée dont la probabilité de faire pile est $1/3$.
- Nous lançons la pièce jusqu'à obtenir face.

Q1 : Quelle est la probabilité de faire $1, 2, 3, 4$ lancés.

Loi géométrique

- Considérons une pièce biaisée dont la probabilité de faire pile est $1/3$.
- Nous lançons la pièce jusqu'à obtenir face.

Q1 : Quelle est la probabilité de faire $1, 2, 3, 4$ lancés.

Réponse : on pose X la v.a. égale au nombre de lancés.

$$\mathbb{P}[X = 1] = \mathbb{P}[face] = \frac{2}{3}, \qquad \mathbb{P}[X = 2] = \mathbb{P}[pile - face] = \frac{1}{3} \times \frac{2}{3},$$

Loi géométrique

- Considérons une pièce biaisée dont la probabilité de faire pile est $1/3$.
- Nous lançons la pièce jusqu'à obtenir face.

Q1 : Quelle est la probabilité de faire $1, 2, 3, 4$ lancés.

Réponse : on pose X la v.a. égale au nombre de lancés.

$$\mathbb{P}[X = 1] = \mathbb{P}[face] = \frac{2}{3}, \qquad \mathbb{P}[X = 2] = \mathbb{P}[pile - face] = \frac{1}{3} \times \frac{2}{3},$$

$$\mathbb{P}[X = 3] = \mathbb{P}[pile - pile - face] = \frac{1}{3} \times \frac{1}{3} \times \frac{2}{3} = \left(\frac{1}{3}\right)^2 \cdot \frac{2}{3},$$

Loi géométrique

- Considérons une pièce biaisée dont la probabilité de faire pile est $1/3$.
- Nous lançons la pièce jusqu'à obtenir face.

Q1 : Quelle est la probabilité de faire $1, 2, 3, 4$ lancés.

Réponse : on pose X la v.a. égale au nombre de lancés.

$$\mathbb{P}[X = 1] = \mathbb{P}[face] = \frac{2}{3}, \qquad \mathbb{P}[X = 2] = \mathbb{P}[pile - face] = \frac{1}{3} \times \frac{2}{3},$$

$$\mathbb{P}[X = 3] = \mathbb{P}[pile - pile - face] = \frac{1}{3} \times \frac{1}{3} \times \frac{2}{3} = \left(\frac{1}{3}\right)^2 \cdot \frac{2}{3},$$

$$\mathbb{P}[X = 4] = \mathbb{P}[pile - pile - pile - face] = \frac{1}{3} \times \frac{1}{3} \times \frac{1}{3} \times \frac{2}{3} = \left(\frac{1}{3}\right)^3 \cdot \frac{2}{3}$$

Loi géométrique

UNIVERSITÉ CAEN NORMANDIE

Loi géométrique

Une variable aléatoire X suit une loi géométrique de paramètres $p \in [0,1]$ si

$$\forall k \geq 1, \qquad \mathbb{P}[X = k] = p(1-p)^{k-1}.$$

L'espérance et la variance de X vérifient (pour $p \neq 0$)

$$\mathbb{E}[X] = \frac{1}{p}, \qquad \mathbb{V}(X) = \frac{1-p}{p^2}.$$

Application : soit x un entier aléatoire de n bits. La probabilité que x soit premier est

$$\mathbb{P}[x \text{ est premier}] = \frac{1}{n \ln 2}.$$

Loi géométrique

UNIVERSITÉ CAEN NORMANDIE

Loi géométrique

Une variable aléatoire X suit une loi géométrique de paramètres $p \in [0,1]$ si

$$\forall k \geq 1, \qquad \mathbb{P}[X = k] = p(1-p)^{k-1}.$$

L'espérance et la variance de X vérifient (pour $p \neq 0$)

$$\mathbb{E}[X] = \frac{1}{p}, \qquad \mathbb{V}(X) = \frac{1-p}{p^2}.$$

Application : soit x un entier aléatoire de n bits. La probabilité que x soit premier est

$$\mathbb{P}[x \text{ est premier}] = \frac{1}{n \ln 2}.$$

On tire x jusqu'à obtenir un nombre premier. Quelle est la probabilité de faire 10 tirages ?

Loi géométrique

UNIVERSITÉ CAEN NORMANDIE

Loi géométrique

Une variable aléatoire X suit une loi géométrique de paramètres $p \in [0,1]$ si

$$\forall k \geq 1, \qquad \mathbb{P}[X = k] = p(1-p)^{k-1}.$$

L'espérance et la variance de X vérifient (pour $p \neq 0$)

$$\mathbb{E}[X] = \frac{1}{p}, \qquad \mathbb{V}(X) = \frac{1-p}{p^2}.$$

Application : soit x un entier aléatoire de n bits. La probabilité que x soit premier est

$$\mathbb{P}[x \text{ est premier}] = \frac{1}{n \ln 2}.$$

On tire x jusqu'à obtenir un nombre premier. Quelle est la probabilité de faire 10 tirages ? Quel est le nombre moyen de tirages ?

Inégalité de Markov

Soit X une v.a. prenant des valeurs positives dans un ensemble E. Soit a un réel strictement positif.

$$\mathbb{P}(X \geq a) \leq \frac{\mathbb{E}[X]}{a}.$$

Inégalité de Markov

Soit X une v.a. prenant des valeurs positives dans un ensemble E. Soit a un réel strictement positif.

$$\mathbb{P}(X \geq a) \leq \frac{\mathbb{E}[X]}{a}.$$

Preuve :

$$\begin{aligned}
\mathbb{E}[X] &= \sum_{x \in E} x\, \mathbb{P}(X = x) \\
&= \sum_{x \in E, x < a} x\, \mathbb{P}(X = x) + \sum_{x \in E, x \geq a} x\, \mathbb{P}(X = x) \\
&\geq \sum_{x \in E, x < a} x\, \mathbb{P}(X = x) \\
&\geq a \sum_{x \in E, x < a} x\, \mathbb{P}(X = x) \\
&\geq a\mathbb{P}(X \geq a)
\end{aligned}$$

Inégalité de Markov

Soit X une v.a. prenant des valeurs positives dans un ensemble E. Soit a un réel strictement positif.

$$\mathbb{P}(X \geq a) \leq \frac{\mathbb{E}[X]}{a}.$$

Preuve :

$$\begin{array}{rcl}
\mathbb{E}[X] & = & \sum_{x \in E} x\, \mathbb{P}(X = x) \\
& = & \sum_{x \in E, x<a} x\, \mathbb{P}(X = x) + \sum_{x \in E, x \geq a} x\, \mathbb{P}(X = x) \\
& \geq & \sum_{x \in E, x<a} x\, \mathbb{P}(X = x) \\
& \geq & a \sum_{x \in E, x<a} x\, \mathbb{P}(X = x) \\
& \geq & a\mathbb{P}(X \geq a)
\end{array}$$

Remarque : si $a >> \mathbb{E}[X]$, alors la probabilité tend vers 0. Il est donc peu probable que X soit d'ordre plus grand que son espérance

Exemple : si $\mathbb{E}[X] \underset{n \to \infty}{\sim} n$, $\quad \mathbb{P}[X > n \ln\ln\ln n] \leq \frac{1}{\ln\ln\ln n} \underset{n \to \infty}{\to} 0$

Inégalité de Bienaymé-Chebychev

Soit X une variable aléatoire. Pour tout $a > 0$,

$$\mathbb{P}\left(|X - \mathbb{E}[X]| \geq a\right) \leq \frac{\mathbb{V}(X)}{a^2}.$$

Inégalité de Bienaymé-Chebychev

Soit X une variable aléatoire. Pour tout $a > 0$,

$$\mathbb{P}\left(|X - \mathbb{E}[X]| \geq a\right) \leq \frac{\mathbb{V}(X)}{a^2}.$$

Preuve :

Comme $Y = (X - \mathbb{E}[X])^2$ est une v.a. prenant des valeurs positives ou nulles, on peut appliquer l'inégalité de Markov sur Y.

$$\mathbb{P}((X - \mathbb{E}[X])^2 \geq a^2) = \frac{\mathbb{E}[(X - \mathbb{E}[X])^2}{a^2} = \frac{\mathbb{V}(X)}{a^2}.$$

Inégalité de Bienaymé-Chebychev

Soit X une variable aléatoire. Pour tout $a > 0$,

$$\mathbb{P}\left(|X - \mathbb{E}[X]| \geq a\right) \leq \frac{\mathbb{V}(X)}{a^2}.$$

Preuve :
Comme $Y = (X - \mathbb{E}[X])^2$ est une v.a. prenant des valeurs positives ou nulles, on peut appliquer l'inégalité de Markov sur Y.

$$\mathbb{P}((X - \mathbb{E}[X])^2 \geq a^2) = \frac{\mathbb{E}[(X - \mathbb{E}[X])^2}{a^2} = \frac{\mathbb{V}(X)}{a^2}.$$

Remarque : si $a^2 >> \mathbb{V}(X)$, alors la probabilité tend vers 0. Il est donc peu probable que X s'écarte de son espérance d'un ordre plus grand que sa variance

Exemple : si $\mathbb{V}(X) \underset{n\to\infty}{\sim} n$, $\quad \mathbb{P}\left(|X - \mathbb{E}[X]| \geq \sqrt{n \ln\ln\ln n}\right) \leq \frac{1}{\ln\ln\ln n} \underset{n\to\infty}{\to} 0$

Loi des grands nombres

UNIVERSITÉ CAEN NORMANDIE

Loi des grands nombres

Soit $X_1, \ldots, X_n$ des variables aléatoires indépendantes de même espérance (notée $\mathbb{E}(X)$) et de même variance (notée $\mathbb{V}(X)$). Alors,

$$\forall \epsilon > 0, \qquad \mathbb{P}\left[\left|\frac{X_1 + X_2 + \cdots + X_n}{n} - \mathbb{E}[X]\right| > \epsilon\right] \underset{n\to\infty}{\to} 0.$$

- Autrement dit, $(X_1 + X_2 + \cdots + X_n)/n$ est un estimateur convergent de $\mathbb{E}[X]$.
- On peut affiner l'erreur commise en utilisant des résultats de grandes déviations

Loi des grands nombres

UNIVERSITÉ CAEN NORMANDIE

- On souhaite estimer une quantité r.

Loi des grands nombres

UNIVERSITÉ CAEN NORMANDIE

- On souhaite estimer une quantité r.
- On construit une expérience aléatoire X telle que $\mathbb{E}[X] = r$

Loi des grands nombres

UNIVERSITÉ CAEN NORMANDIE

- On souhaite estimer une quantité r.
- On construit une expérience aléatoire X telle que $\mathbb{E}[X] = r$
- On répète n fois l'expérience aléatoires X : $X_1, X_2, \ldots, X_n$

Loi des grands nombres

UNIVERSITÉ CAEN NORMANDIE

- On souhaite estimer une quantité r.
- On construit une expérience aléatoire X telle que $\mathbb{E}[X] = r$
- On répète n fois l'expérience aléatoires X : $X_1, X_2, \ldots, X_n$
- Pour n assez grand, la loi des grands nombres donne

$$\frac{X_1 + X_2 + \cdots + X_n}{n} \approx \mathbb{E}[X] = r \qquad \text{au sens probabiliste}$$

Loi des grands nombres

UNIVERSITÉ CAEN NORMANDIE

- On souhaite estimer une quantité r.
- On construit une expérience aléatoire X telle que $\mathbb{E}[X] = r$
- On répète n fois l'expérience aléatoires X : $X_1, X_2, \ldots, X_n$
- Pour n assez grand, la loi des grands nombres donne

$$\frac{X_1 + X_2 + \cdots + X_n}{n} \approx \mathbb{E}[X] = r \qquad \text{au sens probabiliste}$$

- Le résultat n'est pas exacte : une erreur (qui dépend de n) est probablement commise

UNIVERSITÉ
CAEN
NORMANDIE

2- Introduction à l'algorithmique probabiliste

Algorithmes de type Monté-Carlo

Algorithmes de Monté-Carlo

UNIVERSITÉ CAEN NORMANDIE

Algorithme de Monté-carlo

Un algorithme de Monté-Carlo est un algorithme probabiliste

1. dont le temps d'exécution dans le pire des cas est déterministe (borne de complexité qui ne dépend pas de l'aléa),
2. dont le résultat peut être erroné selon une probabilité quantifiable,
3. dont on peut réduire l'erreur (en probabilité) en répétant l'algorithme.

Algorithmes de Monté-Carlo

UNIVERSITÉ CAEN NORMANDIE

Algorithme de Monté-carlo

Un algorithme de Monté-Carlo est un algorithme probabiliste

1. dont le temps d'exécution dans le pire des cas est déterministe (borne de complexité qui ne dépend pas de l'aléa),
2. dont le résultat peut être erroné selon une probabilité quantifiable,
3. dont on peut réduire l'erreur (en probabilité) en répétant l'algorithme.

- La distribution de probabilité de la réponse est telle qu'en itérant l'algorithme, un biais vers la bonne solution se produit

Algorithmes de Monté-Carlo

UNIVERSITÉ CAEN NORMANDIE

Algorithme de Monté-carlo

Un algorithme de Monté-Carlo est un algorithme probabiliste

1. dont le temps d'exécution dans le pire des cas est déterministe (borne de complexité qui ne dépend pas de l'aléa),
2. dont le résultat peut être erroné selon une probabilité quantifiable,
3. dont on peut réduire l'erreur (en probabilité) en répétant l'algorithme.

- La distribution de probabilité de la réponse est telle qu'en itérant l'algorithme, un biais vers la bonne solution se produit
- On ne sait pas forcément déterminer si la réponse retournée est correcte ou non !

Algorithmes de Monté-Carlo

UNIVERSITÉ CAEN NORMANDIE

Algorithme de Monté-carlo

Un algorithme de Monté-Carlo est un algorithme probabiliste

1. dont le temps d'exécution dans le pire des cas est déterministe (borne de complexité qui ne dépend pas de l'aléa),
2. dont le résultat peut être erroné selon une probabilité quantifiable,
3. dont on peut réduire l'erreur (en probabilité) en répétant l'algorithme.

Algorithmes de Monté-Carlo

UNIVERSITÉ CAEN NORMANDIE

Algorithme de Monté-carlo

Un algorithme de Monté-Carlo est un algorithme probabiliste

1. dont le temps d'exécution dans le pire des cas est déterministe (borne de complexité qui ne dépend pas de l'aléa),
2. dont le résultat peut être erroné selon une probabilité quantifiable,
3. dont on peut réduire l'erreur (en probabilité) en répétant l'algorithme.

Exemple 1 : considérons un problème de décision (la réponse est oui/non). Considérons l'algorithme qui avec une probabilité 1/3 retourne "oui" et une probabilité 2/3 retourne "non".
Question : Est-ce un algorithme de Monte-Carlo ?

Algorithmes de Monté-Carlo

UNIVERSITÉ CAEN NORMANDIE

Algorithme de Monté-carlo

Un algorithme de Monté-Carlo est un algorithme probabiliste

1. dont le temps d'exécution dans le pire des cas est déterministe (borne de complexité qui ne dépend pas de l'aléa),
2. dont le résultat peut être erroné selon une probabilité quantifiable,
3. dont on peut réduire l'erreur (en probabilité) en répétant l'algorithme.

Exemple 1 : considérons un problème de décision (la réponse est oui/non). Considérons l'algorithme qui avec une probabilité 1/3 retourne "oui" et une probabilité 2/3 retourne "non".
Question : Est-ce un algorithme de Monte-Carlo ?

→ NON ! En répétant l'algorithme, aucune information sur la réponse n'est apportée (grosso modo, 1/3 de "oui" et 2/3 de "non")

Algorithmes de Monté-Carlo

UNIVERSITÉ CAEN NORMANDIE

Algorithme de Monté-carlo

Un algorithme de Monté-Carlo est un algorithme probabiliste

1. dont le temps d'exécution dans le pire des cas est déterministe (borne de complexité qui ne dépend pas de l'aléa),
2. dont le résultat peut être erroné selon une probabilité quantifiable,
3. dont on peut réduire l'erreur (en probabilité) en répétant l'algorithme.

Algorithmes de Monté-Carlo

UNIVERSITÉ CAEN NORMANDIE

Algorithme de Monté-carlo

Un algorithme de Monté-Carlo est un algorithme probabiliste

1. dont le temps d'exécution dans le pire des cas est déterministe (borne de complexité qui ne dépend pas de l'aléa),
2. dont le résultat peut être erroné selon une probabilité quantifiable,
3. dont on peut réduire l'erreur (en probabilité) en répétant l'algorithme.

Exemple 2 : considérons un problème de décision (la réponse est oui/non). Considérons l'algorithme qui avec une probabilité 1/2 retourne la bonne solution.
Question : Est-ce un algorithme de Monte-Carlo ?

Algorithmes de Monté-Carlo

UNIVERSITÉ CAEN NORMANDIE

Algorithme de Monté-carlo

Un algorithme de Monté-Carlo est un algorithme probabiliste

1. dont le temps d'exécution dans le pire des cas est déterministe (borne de complexité qui ne dépend pas de l'aléa),
2. dont le résultat peut être erroné selon une probabilité quantifiable,
3. dont on peut réduire l'erreur (en probabilité) en répétant l'algorithme.

Exemple 2 : considérons un problème de décision (la réponse est oui/non). Considérons l'algorithme qui avec une probabilité 1/2 retourne la bonne solution.

Question : Est-ce un algorithme de Monte-Carlo ?

→ NON ! En répétant l'algorithme, aucune information sur la réponse n'est apportée (grosso modo, 1/2 de "oui" et 1/2 de "non")

Algorithmes de Monté-Carlo

UNIVERSITÉ CAEN NORMANDIE

Algorithme de Monté-carlo

Un algorithme de Monté-Carlo est un algorithme probabiliste

1. dont le temps d'exécution dans le pire des cas est déterministe (borne de complexité qui ne dépend pas de l'aléa),
2. dont le résultat peut être erroné selon une probabilité quantifiable,
3. dont on peut réduire l'erreur (en probabilité) en répétant l'algorithme.

Algorithmes de Monté-Carlo

UNIVERSITÉ CAEN NORMANDIE

Algorithme de Monté-carlo

Un algorithme de Monté-Carlo est un algorithme probabiliste

1. dont le temps d'exécution dans le pire des cas est déterministe (borne de complexité qui ne dépend pas de l'aléa),
2. dont le résultat peut être erroné selon une probabilité quantifiable,
3. dont on peut réduire l'erreur (en probabilité) en répétant l'algorithme.

Exemple 3 : considérons un problème de décision et un algorithme qui avec une probabilité $\frac{1}{2} + \epsilon$ retourne la bonne solution.
Question : Est-ce un algorithme de Monte-Carlo ?

Algorithmes de Monté-Carlo

UNIVERSITÉ CAEN NORMANDIE

Algorithme de Monté-carlo

Un algorithme de Monté-Carlo est un algorithme probabiliste

1. dont le temps d'exécution dans le pire des cas est déterministe (borne de complexité qui ne dépend pas de l'aléa),
2. dont le résultat peut être erroné selon une probabilité quantifiable,
3. dont on peut réduire l'erreur (en probabilité) en répétant l'algorithme.

Exemple 3 : considérons un problème de décision et un algorithme qui avec une probabilité $\frac{1}{2} + \epsilon$ retourne la bonne solution.
Question : Est-ce un algorithme de Monte-Carlo ?

→ OUI ! En répétant l'algorithme, la bonne réponse devient majoritaire (grosso modo, $\frac{1}{2} + \epsilon$ de réponses correctes et $\frac{1}{2} - \epsilon$ d'incorrectes)

Algorithmes de Monté-Carlo

UNIVERSITÉ CAEN NORMANDIE

Algorithme de Monté-carlo

Un algorithme de Monté-Carlo est un algorithme probabiliste

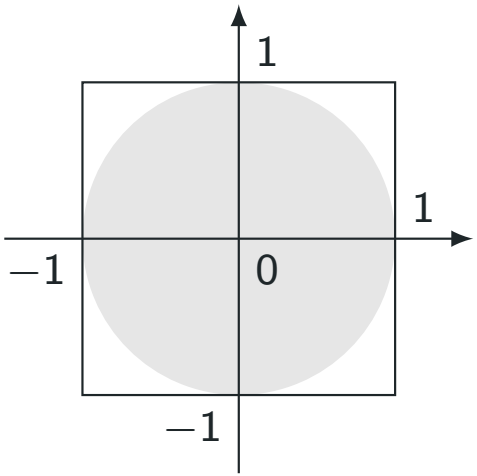
1. dont le temps d'exécution dans le pire des cas est déterministe (borne de complexité qui ne dépend pas de l'aléa),
2. dont le résultat peut être erroné selon une probabilité quantifiable,
3. dont on peut réduire l'erreur (en probabilité) en répétant l'algorithme.

Exemple 3 : considérons un problème de décision et un algorithme qui avec une probabilité $\frac{1}{2} + \epsilon$ retourne la bonne solution.
Question : Est-ce un algorithme de Monte-Carlo ?

→ OUI ! En répétant l'algorithme, la bonne réponse devient majoritaire (grosso modo, $\frac{1}{2} + \epsilon$ de réponses correctes et $\frac{1}{2} - \epsilon$ d'incorrectes)

→ Le ϵ est fondamental (biais)

Estimation de π

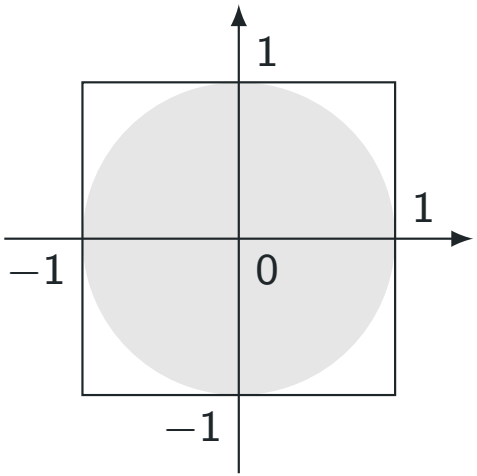


- L'aire du carré est $\mathcal{A}(\text{carré}) = 4$
- L'aire du cercle est $\mathcal{A}(\text{cercle}) = \pi$
- On tire uniformément (x, y) dans le carré
- On pose X la variable aléatoire

$$X = \begin{cases} 1 & \text{si } (x, y) \text{ est dans cercle} \\ 0 & \text{sinon} \end{cases}$$

- loi de X :

Estimation de π



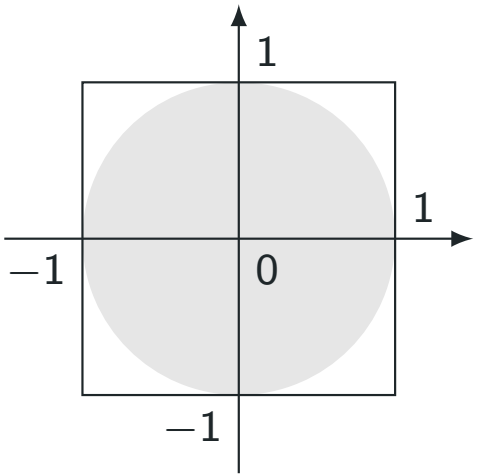
- L'aire du carré est $\mathcal{A}(\text{carré}) = 4$
- L'aire du cercle est $\mathcal{A}(\text{cercle}) = \pi$
- On tire uniformément (x, y) dans le carré
- On pose X la variable aléatoire

$$X = \begin{cases} 1 & \text{si } (x,y) \text{ est dans cercle} \\ 0 & \text{sinon} \end{cases}$$

- loi de X : $X \sim \text{Bern}\left(\frac{\pi}{4}\right)$

$$\mathbb{P}[X = 1] = \frac{\pi}{4}, \qquad \mathbb{P}[X = 0] = 1 - \frac{\pi}{4}, \quad \mathbb{E}[X] = \frac{\pi}{4}.$$

Estimation de π



- L'aire du carré est $\mathcal{A}(\text{carré}) = 4$
- L'aire du cercle est $\mathcal{A}(\text{cercle}) = \pi$
- On tire uniformément (x, y) dans le carré
- On pose X la variable aléatoire

$$X = \begin{cases} 1 & \text{si } (x,y) \text{ est dans cercle} \\ 0 & \text{sinon} \end{cases}$$

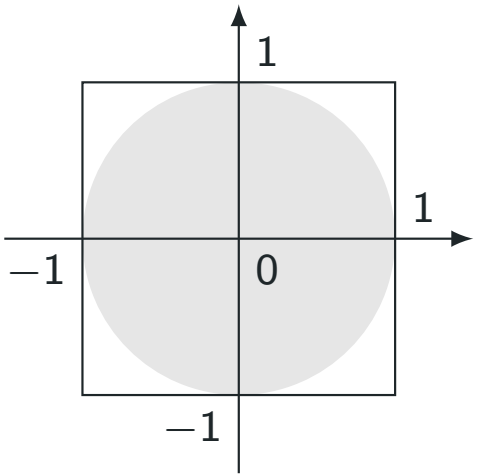
- loi de X : $X \sim \text{Bern}\left(\frac{\pi}{4}\right)$

$$\mathbb{P}[X = 1] = \frac{\pi}{4}, \qquad \mathbb{P}[X = 0] = 1 - \frac{\pi}{4}, \quad \mathbb{E}[X] = \frac{\pi}{4}.$$

- Loi des grands nombres : si $X_1, \ldots, X_n$ sont i.i.d de même loi $\text{Bern}\left(\frac{\pi}{4}\right)$ alors

$$\frac{X_1 + \cdots + X_n}{n} \xrightarrow[n\to\infty]{}$$

Estimation de π



- L'aire du carré est $\mathcal{A}(\text{carré}) = 4$
- L'aire du cercle est $\mathcal{A}(\text{cercle}) = \pi$
- On tire uniformément (x, y) dans le carré
- On pose X la variable aléatoire

$$X = \begin{cases} 1 & \text{si } (x,y) \text{ est dans cercle} \\ 0 & \text{sinon} \end{cases}$$

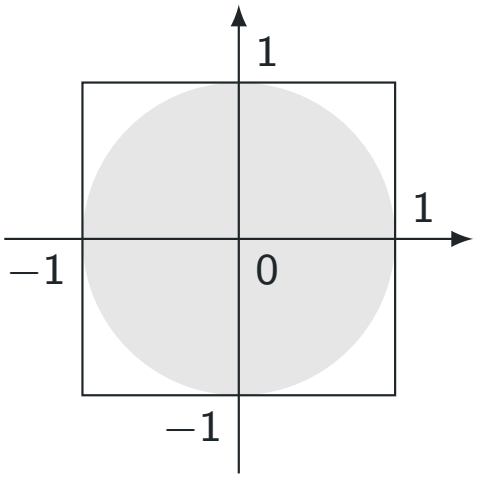
- loi de X : $X \sim \text{Bern}\left(\frac{\pi}{4}\right)$

$$\mathbb{P}[X = 1] = \frac{\pi}{4}, \qquad \mathbb{P}[X = 0] = 1 - \frac{\pi}{4}, \quad \mathbb{E}[X] = \frac{\pi}{4}.$$

- Loi des grands nombres : si $X_1, \ldots, X_n$ sont i.i.d de même loi $\text{Bern}\left(\frac{\pi}{4}\right)$ alors

$$\frac{X_1 + \cdots + X_n}{n} \xrightarrow[n\to\infty]{} \frac{\pi}{4} \qquad p.s. \qquad \text{(plus de détails en TD)}$$

Estimation de π



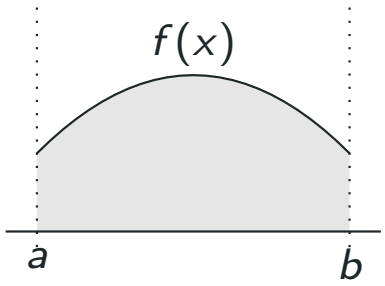
Estimation de π

Entrée : n le nombre de tests
Sortie : Une estimation de π après n tests

1 : $compteur = 0$
2 : **Pour** $i = 1$ **à** n **do**
3 : choisir x et y uniformément dans $[-1, 1]$
4 : **Si** $x^2 + y^2 \leq 1$ **Alors**
5 : incrémenter *compteur* de 1
6 : **Fin Si**
7 : **Fin Pour**
8 : **Retourner** $compteur \times 4/n$

- Nous verrons en TD comment choisir n pour une précision donnée avec une probabilité fixée.

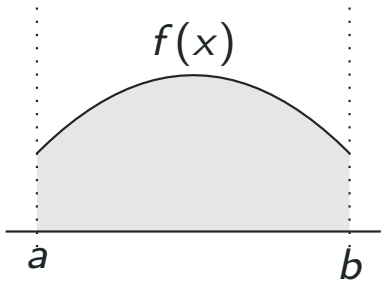
Estimation d'une intégrale



- On pose X la variable aléatoire uniforme sur $[a, b]$
- On pose Y la variable aléatoire $Y = f(X)$.
- Nous avons

$$\mathbb{E}[Y] =$$

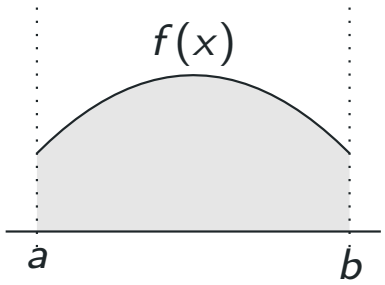
Estimation d'une intégrale



- On pose X la variable aléatoire uniforme sur $[a, b]$
- On pose Y la variable aléatoire $Y = f(X)$.
- Nous avons

$$\mathbb{E}[Y] = \frac{1}{b-a} \int_a^b f(x)dx.$$

Estimation d'une intégrale



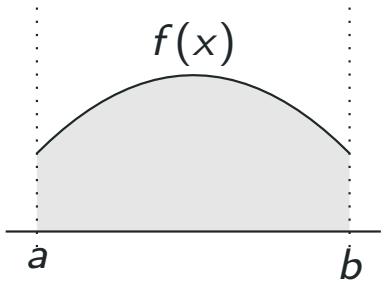
- On pose X la variable aléatoire uniforme sur $[a, b]$
- On pose Y la variable aléatoire $Y = f(X)$.
- Nous avons

$$\mathbb{E}[Y] = \frac{1}{b-a} \int_a^b f(x)dx.$$

- Loi des grands nombres : soit $X_1, \ldots, X_n$ i.i.d de même loi uniforme sur $[a, b]$. On pose $Y_i = f(X_i)$ pour $i = 1 \ldots n$.Alors

$$\frac{Y_1 + \cdots + Y_n}{n} \xrightarrow[n \to \infty]{}$$

Estimation d'une intégrale



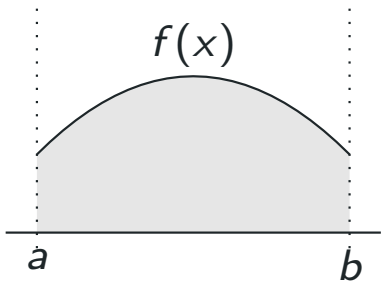
- On pose X la variable aléatoire uniforme sur $[a, b]$
- On pose Y la variable aléatoire $Y = f(X)$.
- Nous avons

$$\mathbb{E}[Y] = \frac{1}{b-a} \int_a^b f(x)dx.$$

- Loi des grands nombres : soit $X_1, \ldots, X_n$ i.i.d de même loi uniforme sur $[a, b]$. On pose $Y_i = f(X_i)$ pour $i = 1 \ldots n$.Alors

$$\frac{Y_1 + \cdots + Y_n}{n} \underset{n\to\infty}{\longrightarrow} \mathbb{E}[Y] = \frac{1}{b-a} \int_a^b f(x)dx.$$

Estimation d'une intégrale



Estimation d'une intégrale

Entrée : une fonction f sur $[a, b]$,
n le nombre de points dans l'intervalle $[a, b]$

Sortie : Une estimation de $\int_a^b f(x)dx$.

1 : $accu = 0$
2 : **Pour** $i = 1$ **à** n **do**
3 : choisir x $[a, b]$
4 : ajouter $f(x)$ à $accu$
5 : **Fin Pour**
6 : **Retourner** $(b - a) \times accu/n$

- Nous verrons en TD comment choisir n pour une précision donnée avec une probabilité fixée.
- L'algorithme fonctionne aussi pour les intégrales multiples !

Élection d'un leader dans un réseau

Un réseau est modélisé par un graphe complet $G = (V, E)$ à n sommets.

On souhaite élire un leader dans le réseau (algorithme distribué)

Élection d'un leader dans un réseau

Un réseau est modélisé par un graphe complet $G = (V, E)$ à n sommets.

On souhaite élire un leader dans le réseau (algorithme distribué)

Élection d'un leader

Entrée : Partager un entier N avec ses voisins

1 : Chaque sommet v tire un entier $r(v)$ uniformément dans $[1, N]$

2 : Chaque sommet v transmet $r(v)$ à ses voisins

3 : Pour un sommet v, **Si** $r(v) > r(v')$ pour tous ses voisins v'

Alors $Etat_v = elu$

Sinon $Etat_v = non - elu$

4 : Chaque sommet transmet son état à ses voisins et reçoit l'état de ses voisins

Élection d'un leader dans un réseau

Un réseau est modélisé par un graphe complet $G = (V, E)$ à n sommets.

On souhaite élire un leader dans le réseau (algorithme distribué)

Élection d'un leader

Entrée : Partager un entier N avec ses voisins

1 : Chaque sommet v tire un entier $r(v)$ uniformément dans $[1, N]$

2 : Chaque sommet v transmet $r(v)$ à ses voisins

3 : Pour un sommet v, **Si** $r(v) > r(v')$ pour tous ses voisins v'
Alors $Etat_v = elu$
Sinon $Etat_v = non - elu$

4 : Chaque sommet transmet son état à ses voisins et reçoit l'état de ses voisins

- L'algorithme fonctionne-t'il à tous les coups ?

Élection d'un leader dans un réseau

Un réseau est modélisé par un graphe complet $G = (V, E)$ à n sommets.

On souhaite élire un leader dans le réseau (algorithme distribué)

Élection d'un leader

Entrée : Partager un entier N avec ses voisins

1 : Chaque sommet v tire un entier $r(v)$ uniformément dans $[1, N]$

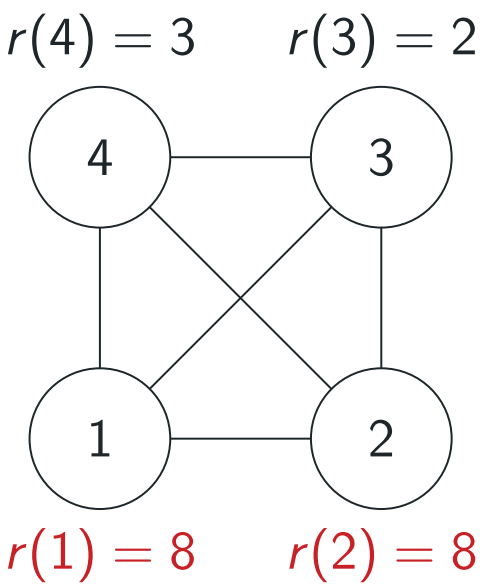
2 : Chaque sommet v transmet $r(v)$ à ses voisins

3 : Pour un sommet v, **Si** $r(v) > r(v')$ pour tous ses voisins v'
Alors $Etat_v = elu$
Sinon $Etat_v = non - elu$

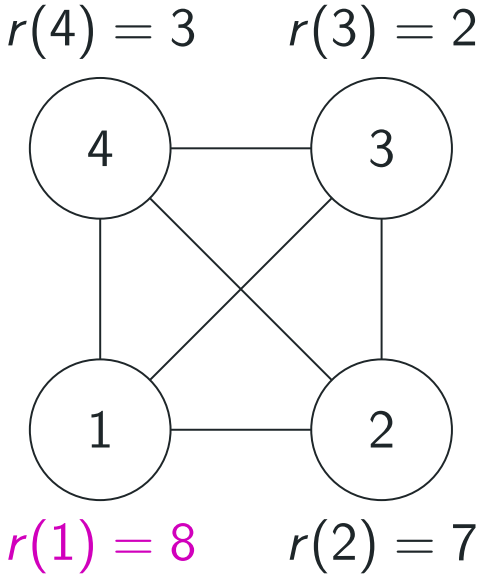
4 : Chaque sommet transmet son état à ses voisins et reçoit l'état de ses voisins

- L'algorithme fonctionne-t'il à tous les coups ? Non !

Élection d'un leader dans un réseau



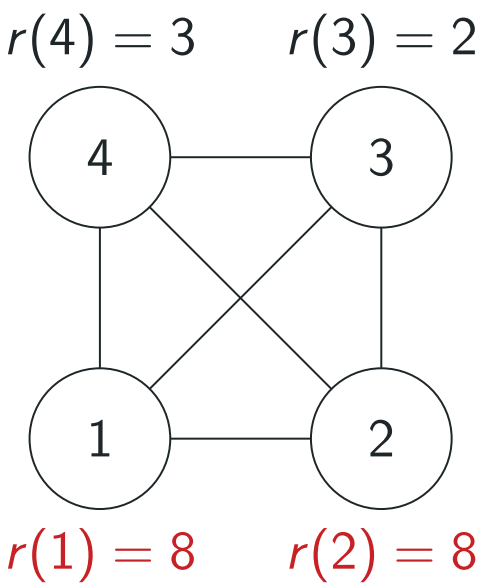
Pas d'élection !



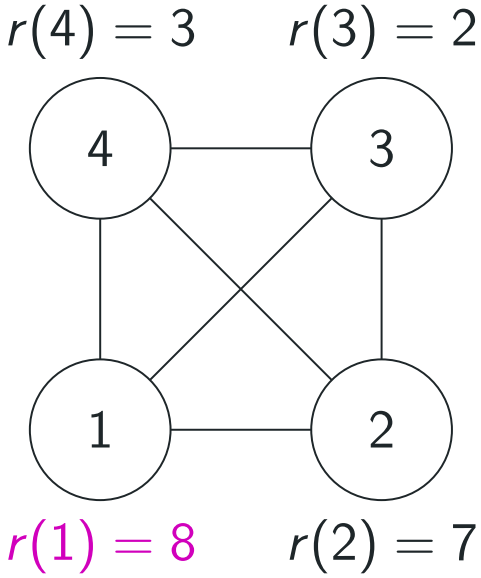
Élection

- Quelle est la probabilité de succès de l'algorithme (pour $N = n$) ?

Élection d'un leader dans un réseau



Pas d'élection !

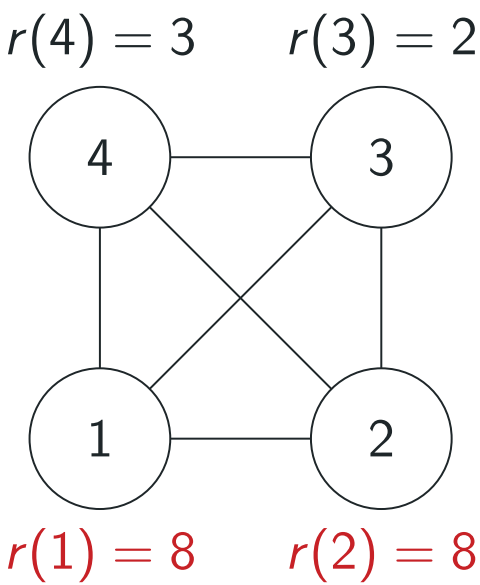


Élection

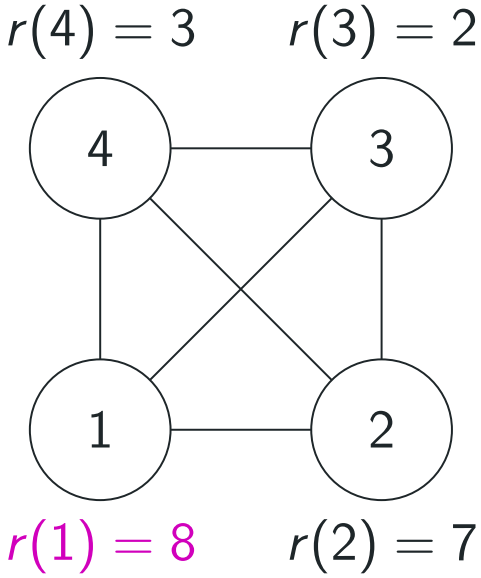
- Quelle est la probabilité de succès de l'algorithme (pour $N = n$) ?

$$\mathbb{P}[succes] = n \times \sum_{i=2}^{n} \frac{1}{n} \cdot \left(\frac{i-1}{n}\right)^{n-1} \sim \frac{e^{-1}}{1-e^{-1}} \approx 0,58$$

Élection d'un leader dans un réseau



Pas d'élection !



Élection

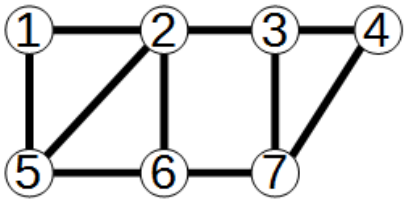
- Quelle est la probabilité de succès de l'algorithme (pour $N = n$) ?

$$\mathbb{P}[succes] = n \times \sum_{i=2}^{n} \frac{1}{n} \cdot \left(\frac{i-1}{n}\right)^{n-1} \sim \frac{e^{-1}}{1-e^{-1}} \approx 0,58$$

- En augmentant N, la probabilité tend vers 1

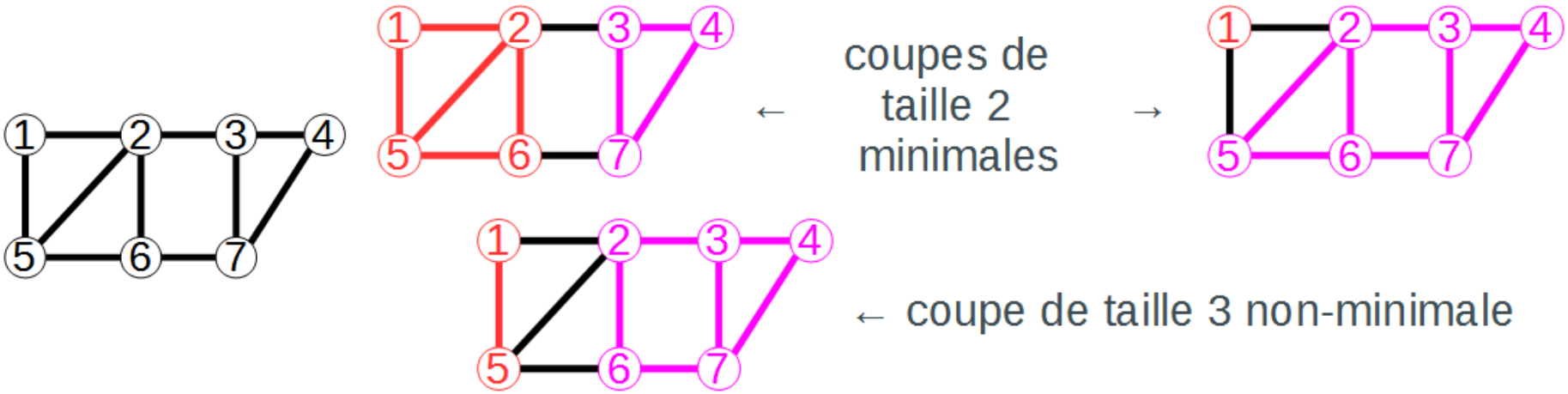
Algorithme de Karger et coupe minimale

- Considérons un graphe $G = (V, E)$ à n sommets.
- Une coupe du graphe G est une partition des sommets en deux sous-ensembles $V = V_1 \cup V_2$ $(V_1 \cap V_2 = \emptyset)$
- La taille d'une coupe $V = V_1 \cup V_2$ est le nombre d'arêtes entre un sommet de V_1 et un sommet de V_2.
- Ex : donnez une coupe minimale (de plus petite taille) du graphe suivant :



Algorithme de Karger et coupe minimale

- Considérons un graphe $G = (V, E)$ à n sommets.
- Une coupe du graphe G est une partition des sommets en deux sous-ensembles $V = V_1 \cup V_2$ $(V_1 \cap V_2 = \emptyset)$
- La taille d'une coupe $V = V_1 \cup V_2$ est le nombre d'arêtes entre un sommet de V_1 et un sommet de V_2.
- Ex : donnez une coupe minimale (de plus petite taille) du graphe suivant :



Algorithme de Karger et coupe minimale

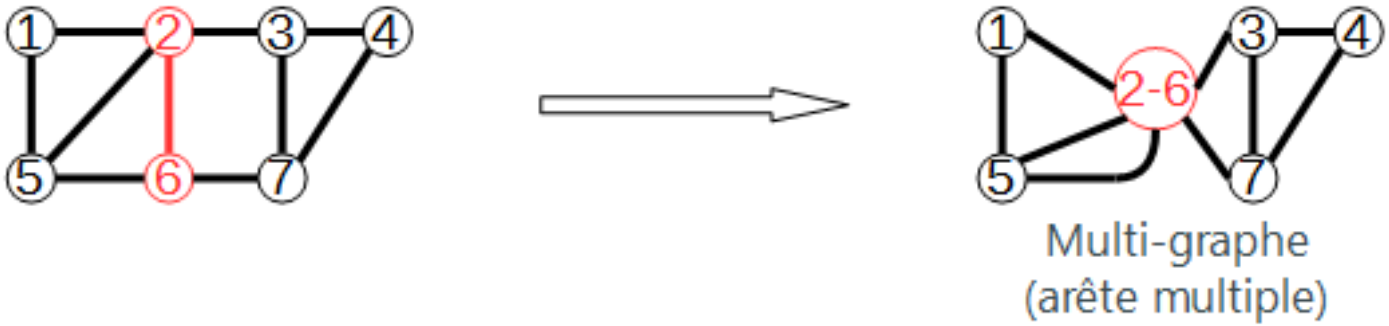
UNIVERSITÉ CAEN NORMANDIE

Problème MIN-CUT : étant donné un graphe $G = (V, E)$, trouver une coupe minimale de G.

Algorithme de Karger et coupe minimale

Problème MIN-CUT : étant donné un graphe $G = (V, E)$, trouver une coupe minimale de G.

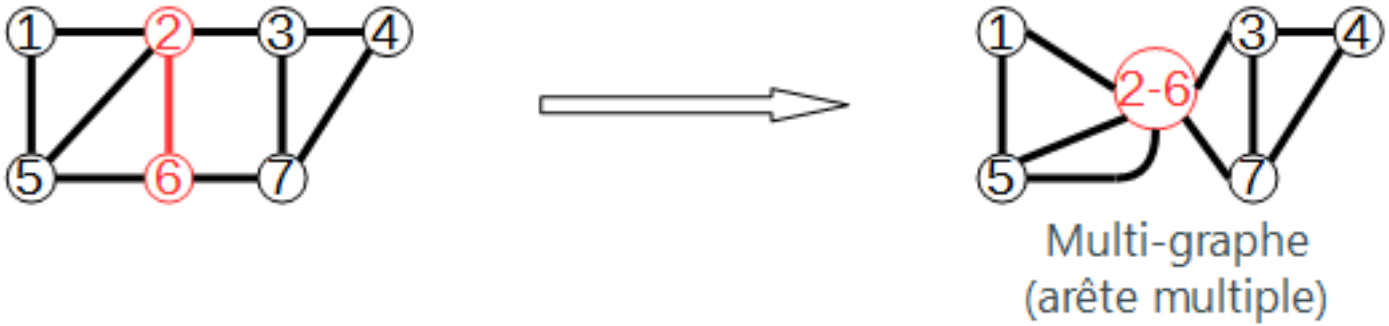
- Contraction d'une arête (fusion de deux sommets) :



Algorithme de Karger et coupe minimale

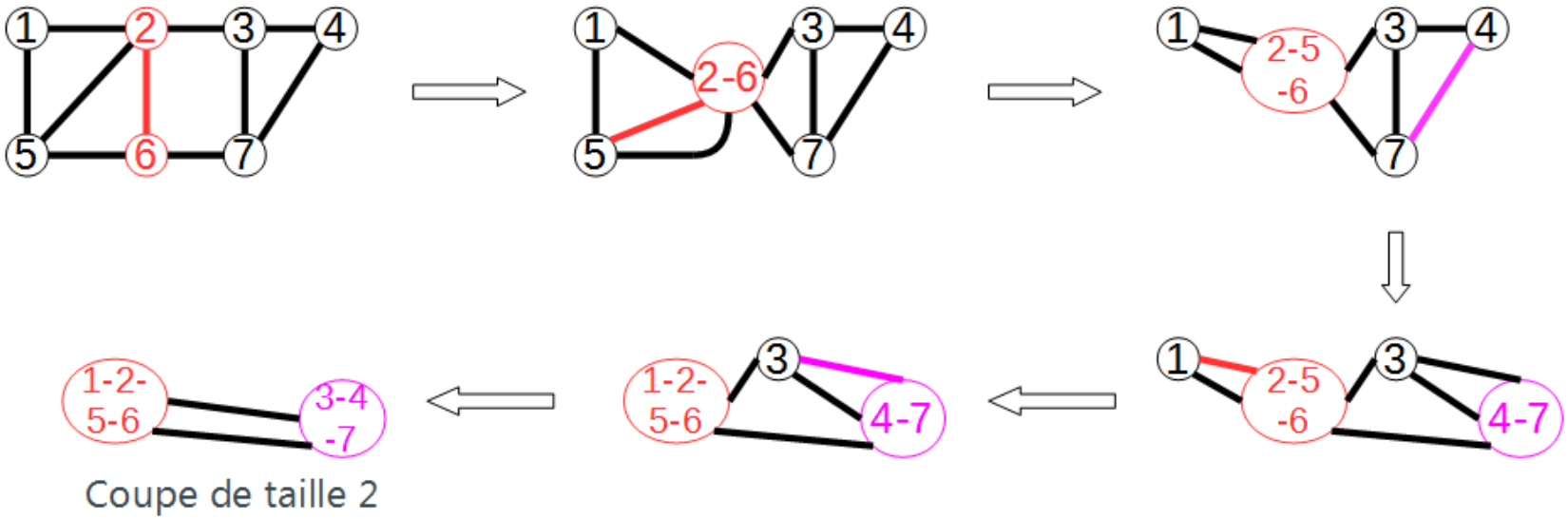
Problème MIN-CUT : étant donné un graphe $G = (V, E)$, trouver une coupe minimale de G.

- Contraction d'une arête (fusion de deux sommets) :

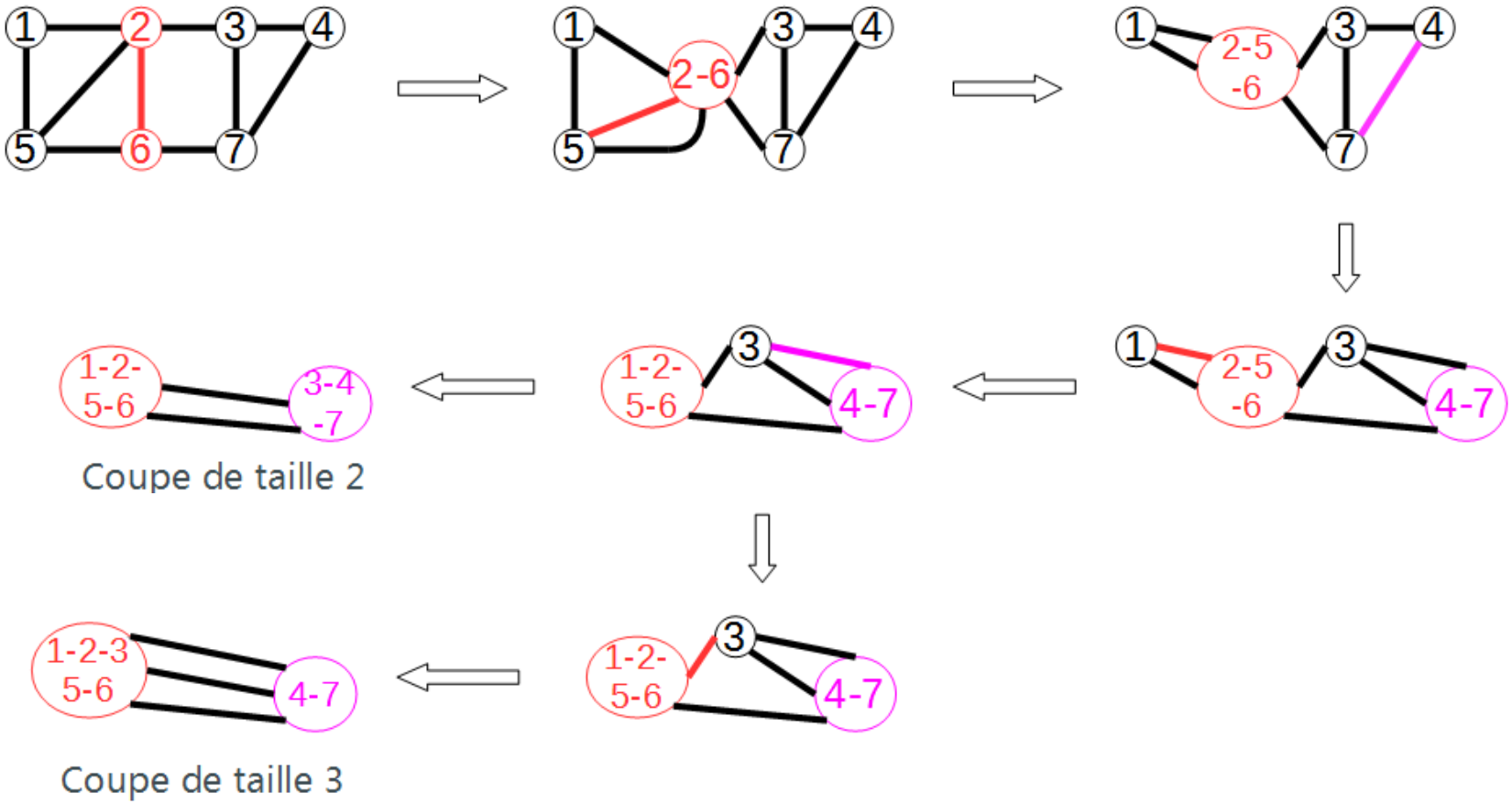


- Algorithme de Karger
  - Tant qu'une contraction est possible, en choisir une au hasard et contracter
  - A la fin, il ne reste que deux sommets dont les labels forment une coupe
  - Le résultat peut être erroné (algorithme de type Monte-Carlo)

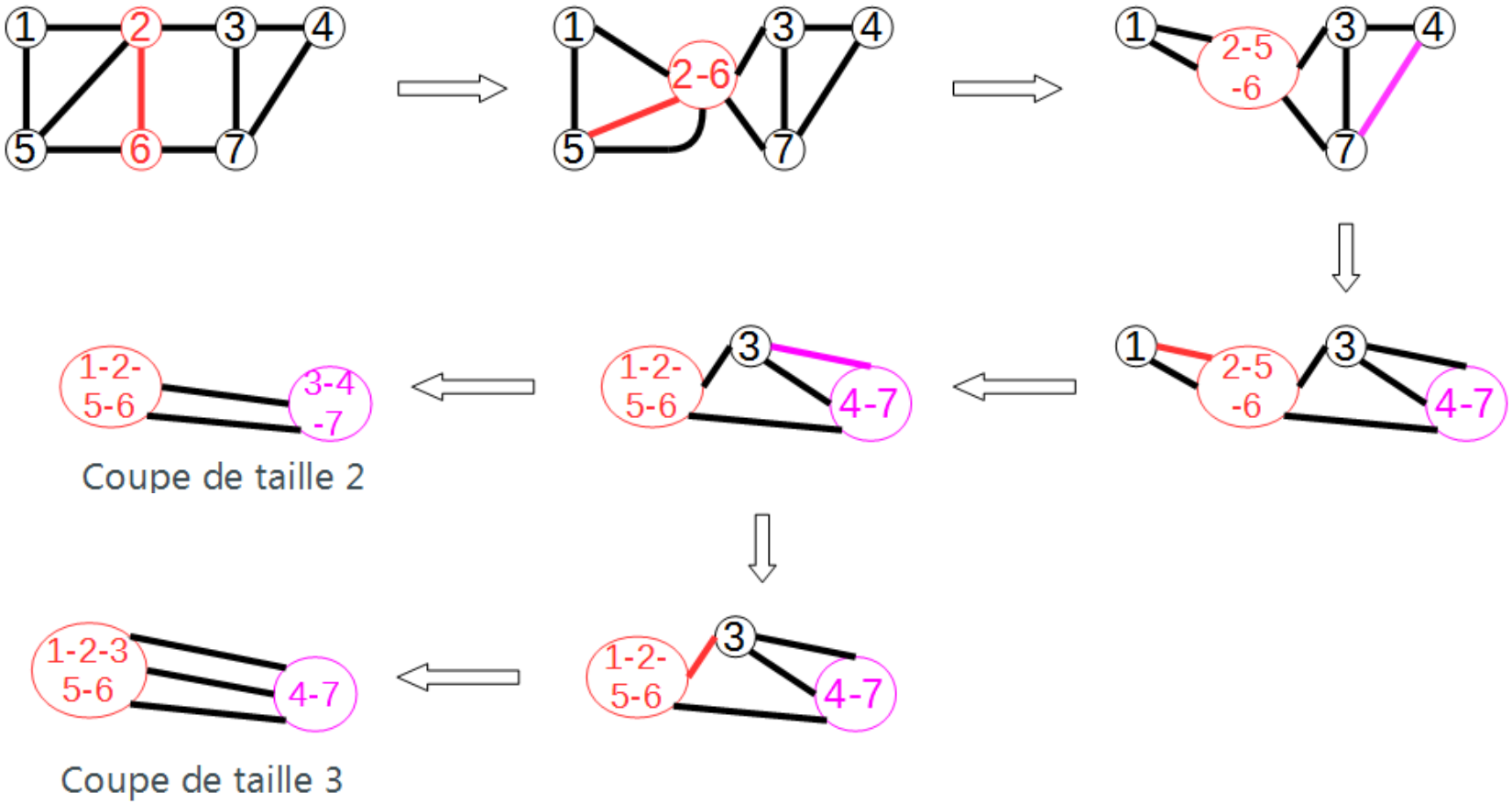
Algorithme de Karger et coupe minimale



Algorithme de Karger et coupe minimale



Algorithme de Karger et coupe minimale



- L'algorithme échoue/réussit à trouver une coupe minimale avec une certaine probabilité
- On ne sait pas lorsque l'algorithme échoue !

Algorithme de Karger et coupe minimale

UNIVERSITÉ CAEN NORMANDIE

Supposons que le (multi-)graphe a n sommets et une coupe minimale de taille t.

- Le nombre d'arêtes du (multi-)graphe est

Algorithme de Karger et coupe minimale

UNIVERSITÉ CAEN NORMANDIE

Supposons que le (multi-)graphe a n sommets et une coupe minimale de taille t.

- Le nombre d'arêtes du (multi-)graphe est $\geq tn/2$.

Algorithme de Karger et coupe minimale

UNIVERSITÉ CAEN NORMANDIE

Supposons que le (multi-)graphe a n sommets et une coupe minimale de taille t.

- Le nombre d'arêtes du (multi-)graphe est $\geq tn/2$.
- La probabilité de choisir une arête de la coupe est $t/(tn/2) = 2/n$.

Algorithme de Karger et coupe minimale

UNIVERSITÉ CAEN NORMANDIE

Supposons que le (multi-)graphe a n sommets et une coupe minimale de taille t.

- Le nombre d'arêtes du (multi-)graphe est $\geq tn/2$.
- La probabilité de choisir une arête de la coupe est $t/(tn/2) = 2/n$.
- La probabilité de ne pas choisir une arête de la coupe est donc $1 - \frac{2}{n}$.

Algorithme de Karger et coupe minimale

UNIVERSITÉ CAEN NORMANDIE

Supposons que le (multi-)graphe a n sommets et une coupe minimale de taille t.

- Le nombre d'arêtes du (multi-)graphe est $\geq tn/2$.
- La probabilité de choisir une arête de la coupe est $t/(tn/2) = 2/n$.
- La probabilité de ne pas choisir une arête de la coupe est donc $1 - \frac{2}{n}$.
- Après la contraction d'une arête, le (multi-)graphe a $n - 1$ sommets et une coupe minimale de taille t (si aucune arête de la coupe n'a été choisie)

Algorithme de Karger et coupe minimale

Supposons que le (multi-)graphe a n sommets et une coupe minimale de taille t.

- Le nombre d'arêtes du (multi-)graphe est $\geq tn/2$.
- La probabilité de choisir une arête de la coupe est $t/(tn/2) = 2/n$.
- La probabilité de ne pas choisir une arête de la coupe est donc $1 - \frac{2}{n}$.
- Après la contraction d'une arête, le (multi-)graphe a $n-1$ sommets et une coupe minimale de taille t (si aucune arête de la coupe n'a été choisie)
- Par récurrence sur n, la probabilité de ne jamais choisir une arête de la coupe minimale est

$$\geq \left(1-\frac{2}{n}\right)\left(1-\frac{2}{n-1}\right)\left(1-\frac{2}{n-2}\right)\cdots\left(1-\frac{2}{3}\right) = \frac{2}{n(n-1)} \geq \frac{2}{n^2}$$

Algorithme de Karger et coupe minimale

UNIVERSITÉ CAEN NORMANDIE

Supposons que le (multi-)graphe a n sommets et une coupe minimale de taille t.

- Le nombre d'arêtes du (multi-)graphe est $\geq tn/2$.
- La probabilité de choisir une arête de la coupe est $t/(tn/2) = 2/n$.
- La probabilité de ne pas choisir une arête de la coupe est donc $1 - \frac{2}{n}$.
- Après la contraction d'une arête, le (multi-)graphe a $n-1$ sommets et une coupe minimale de taille t (si aucune arête de la coupe n'a été choisie)
- Par récurrence sur n, la probabilité de ne jamais choisir une arête de la coupe minimale est

$$\geq \left(1-\frac{2}{n}\right)\left(1-\frac{2}{n-1}\right)\left(1-\frac{2}{n-2}\right)\cdots\left(1-\frac{2}{3}\right) = \frac{2}{n(n-1)} \geq \frac{2}{n^2}$$

- En exécutant $cn^2/2$ fois l'algorithme de Karger, on trouve une coupe minimale avec une probabilité

Algorithme de Karger et coupe minimale

Supposons que le (multi-)graphe a n sommets et une coupe minimale de taille t.

- Le nombre d'arêtes du (multi-)graphe est $\geq tn/2$.
- La probabilité de choisir une arête de la coupe est $t/(tn/2) = 2/n$.
- La probabilité de ne pas choisir une arête de la coupe est donc $1 - \frac{2}{n}$.
- Après la contraction d'une arête, le (multi-)graphe a $n-1$ sommets et une coupe minimale de taille t (si aucune arête de la coupe n'a été choisie)
- Par récurrence sur n, la probabilité de ne jamais choisir une arête de la coupe minimale est

$$\geq \left(1-\frac{2}{n}\right)\left(1-\frac{2}{n-1}\right)\left(1-\frac{2}{n-2}\right)\cdots\left(1-\frac{2}{3}\right) = \frac{2}{n(n-1)} \geq \frac{2}{n^2}$$

- En exécutant $cn^2/2$ fois l'algorithme de Karger, on trouve une coupe minimale avec une probabilité

$$1-\left(1-\frac{2}{n^2}\right)^{cn^2/2} \sim 1-e^{-c} \quad \underset{c\to\infty}{\to} 1$$

Algorithmes de Monté-Carlo et biais

UNIVERSITÉ CAEN NORMANDIE

1. Un algorithme de Monté-Carlo est biaisé vers le *faux* si lorsqu'il répond *faux*, sa réponse est toujours correcte.
2. Un algorithme de Monté-Carlo est biaisé vers le *vrai* si lorsqu'il répond *vrai*, sa réponse est toujours correcte.

Exemples :

- Test de nullité d'un polynôme
- Test de primalité de Miller-Rabin

Test de nullité d'un polynôme

UNIVERSITÉ CAEN NORMANDIE

Soit $P(X)$ un polynôme de degré d sur un corps (fini ou infini) très grand.
Problème : La fonction $x \to P(x)$ est-elle nulle ?

Exemple : la fonction $x \mapsto x^q - x$ est nulle sur le corps fini $\mathbb{F}_q$.

Test de nullité d'un polynôme

UNIVERSITÉ CAEN NORMANDIE

Soit $P(X)$ un polynôme de degré d sur un corps (fini ou infini) très grand.
Problème : La fonction $x \to P(x)$ est-elle nulle ?

Exemple : la fonction $x \mapsto x^q - x$ est nulle sur le corps fini $\mathbb{F}_q$.

Algorithme 1 : Tester toutes les valeurs (complexité : $O(qd)$)

Test de nullité d'un polynôme

UNIVERSITÉ CAEN NORMANDIE

Soit $P(X)$ un polynôme de degré d sur un corps (fini ou infini) très grand.
Problème : La fonction $x \to P(x)$ est-elle nulle ?

Exemple : la fonction $x \mapsto x^q - x$ est nulle sur le corps fini $\mathbb{F}_q$.

Algorithme 1 : Tester toutes les valeurs (complexité : $O(qd)$)

Algorithme 2 : Tester $d+1$ valeurs (complexité : $O(d^2)$) car un polynôme de degré d a au plus d racines

Test de nullité d'un polynôme

UNIVERSITÉ CAEN NORMANDIE

Soit $P(X)$ un polynôme de degré d sur un corps (fini ou infini) très grand.
Problème : La fonction $x \to P(x)$ est-elle nulle ?

Exemple : la fonction $x \mapsto x^q - x$ est nulle sur le corps fini $\mathbb{F}_q$.

Algorithme 1 : Tester toutes les valeurs (complexité : $O(qd)$)

Algorithme 2 : Tester $d+1$ valeurs (complexité : $O(d^2)$) car un polynôme de degré d a au plus d racines

Algorithme 3 : Tester 1 valeur (complexité : $O(d)$) mais la probabilité d'échec est bornée par

Test de nullité d'un polynôme

UNIVERSITÉ CAEN NORMANDIE

Soit $P(X)$ un polynôme de degré d sur un corps (fini ou infini) très grand.
Problème : La fonction $x \to P(x)$ est-elle nulle ?

Exemple : la fonction $x \mapsto x^q - x$ est nulle sur le corps fini $\mathbb{F}_q$.

Algorithme 1 : Tester toutes les valeurs (complexité : $O(qd)$)

Algorithme 2 : Tester $d+1$ valeurs (complexité : $O(d^2)$) car un polynôme de degré d a au plus d racines

Algorithme 3 : Tester 1 valeur (complexité : $O(d)$) mais la probabilité d'échec est bornée par d/q

Test de nullité d'un polynôme

UNIVERSITÉ CAEN NORMANDIE

Soit $P(X)$ un polynôme de degré d sur un corps (fini ou infini) très grand.
Problème : La fonction $x \to P(x)$ est-elle nulle ?

Exemple : la fonction $x \mapsto x^q - x$ est nulle sur le corps fini $\mathbb{F}_q$.

Algorithme 1 : Tester toutes les valeurs (complexité : $O(qd)$)

Algorithme 2 : Tester $d+1$ valeurs (complexité : $O(d^2)$) car un polynôme de degré d a au plus d racines

Algorithme 3 : Tester 1 valeur (complexité : $O(d)$) mais la probabilité d'échec est bornée par d/q

Algorithme 4 : Tester k valeurs aléatoires (complexité : $O(kd)$) mais la probabilité d'échec est bornée par

Test de nullité d'un polynôme

UNIVERSITÉ CAEN NORMANDIE

Soit $P(X)$ un polynôme de degré d sur un corps (fini ou infini) très grand.
Problème : La fonction $x \to P(x)$ est-elle nulle ?

Exemple : la fonction $x \mapsto x^q - x$ est nulle sur le corps fini $\mathbb{F}_q$.

Algorithme 1 : Tester toutes les valeurs (complexité : $O(qd)$)

Algorithme 2 : Tester $d+1$ valeurs (complexité : $O(d^2)$) car un polynôme de degré d a au plus d racines

Algorithme 3 : Tester 1 valeur (complexité : $O(d)$) mais la probabilité d'échec est bornée par d/q
Algorithme 4 : Tester k valeurs aléatoires (complexité : $O(kd)$) mais la probabilité d'échec est bornée par $(d/q)^k$

Test de nullité d'un polynôme

UNIVERSITÉ CAEN NORMANDIE

Soit $P(X)$ un polynôme de degré d sur un corps (fini ou infini) très grand.
Problème : La fonction $x \to P(x)$ est-elle nulle ?

Exemple : la fonction $x \mapsto x^q - x$ est nulle sur le corps fini $\mathbb{F}_q$.

Algorithme 1 : Tester toutes les valeurs (complexité : $O(qd)$)

Algorithme 2 : Tester $d+1$ valeurs (complexité : $O(d^2)$) car un polynôme de degré d a au plus d racines

Algorithme 3 : Tester 1 valeur (complexité : $O(d)$) mais la probabilité d'échec est bornée par d/q
Algorithme 4 : Tester k valeurs aléatoires (complexité : $O(kd)$) mais la probabilité d'échec est bornée par $(d/q)^k$
$\to$ Si $q >> d$, alors l'algorithme a peu de chance de se tromper.

Test de nullité d'un polynôme

UNIVERSITÉ CAEN NORMANDIE

Soit $P(X)$ un polynôme de degré d sur un corps (fini ou infini) très grand.
Problème : La fonction $x \to P(x)$ est-elle nulle ?

Exemple : la fonction $x \mapsto x^q - x$ est nulle sur le corps fini $\mathbb{F}_q$.

Algorithme 1 : Tester toutes les valeurs (complexité : $O(qd)$)

Algorithme 2 : Tester $d+1$ valeurs (complexité : $O(d^2)$) car un polynôme de degré d a au plus d racines

Algorithme 3 : Tester 1 valeur (complexité : $O(d)$) mais la probabilité d'échec est bornée par d/q

Algorithme 4 : Tester k valeurs aléatoires (complexité : $O(kd)$) mais la probabilité d'échec est bornée par $(d/q)^k$

$\to$ Si $q >> d$, alors l'algorithme a peu de chance de se tromper.

Rem : le résultat est le même pour des polynômes multivariés !

Test de nullité d'un polynôme

UNIVERSITÉ CAEN NORMANDIE

Test polynôme nul

Entrée : $P \in \mathbb{F}[x_1, \ldots, x_n]$ un polynôme sur le corps $\mathbb{F}$
$S \subset \mathbb{F}$ un sous-ensemble fini de $\mathbb{F}$
Sortie : Retourne si $x \mapsto P(x)$ est nulle
1 : Choisir $(v_1, \ldots, v_n)$ uniformément dans S
2 : Calculer $r = P(v_1, \ldots, v_n)$
3 : **Si** $r \neq 0$ **Alors**
4 : **Retourner** *non − nul*
5 : **Sinon**
6 : **Retourner** *probablement nul*
7 : **Fin Si**

- C'est un algorithme de Monté-Carlo biaisé vers le *faux* (non-nul)
- La probabilité d'erreur dans l'autre cas est $\leq \frac{d}{\mathrm{Card}(S)}$
- En effectuant au plus k appels au test, la probabilité d'erreur $\leq \left(\frac{d}{\mathrm{Card}(S)}\right)^k$

Test de nullité d'un polynôme

Exemple :

- Posons $P = x_1^2 - x_1 \cdot x_2^2$ sur le corps fini $\mathbb{F}_7 = \mathbb{Z}/7\mathbb{Z}$
- Considérons $S = F$. Nous avons $d = 3$ et $\mathrm{Card}(S) = 7$ soit,

$$\text{si}(v_1, v_2) \in_{\mathcal{R}} S \times S, \qquad \Pr\left[P(v_1, v_2) = 0\right] \leq \frac{d}{\mathrm{Card}(S)} = \frac{3}{7}$$

- Vérification :

x_1 \ x_2	0	1	2	3	4	5	6
0	0	0	0	0	0	0	0
1	1	0	4	6	6	4	0
2	4	2	3	0	0	3	2
3	2	6	4	3	3	4	6
4	2	5	0	1	1	0	5
5	4	6	5	1	1	5	6
6	1	2	5	3	3	5	2

$$\text{si}(v_1, v_2) \in_{\mathcal{R}} S \times S, \qquad \Pr\left[P(v_1, v_2) = 0\right] = \frac{13}{49} \leq \frac{3}{7}$$

Primalité

UNIVERSITÉ CAEN NORMANDIE

- $\{2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97, 101, 103, 107, 109, 113, 127, 131, 137, 139, 149, 151, 157, 163, 167, 173, 179, 181, 191, 193, 197, 199, 211, 223, 227, 229, 233, 239, 241, 251, 257, 263, 269, 271, 277, 281, \ldots\}$

- `http://www.lix.polytechnique.fr/~morain/Primes/myprimes.html`
- `http://primes.utm.edu/`

Primalité

UNIVERSITÉ CAEN NORMANDIE

- Il existe une infinité de nombres premiers
- $\pi(x) = \#\{n \leq x : n \text{ est premier}\}$

  On a

  $$\pi(x) \sim x/\ln x$$

- Si on tire uniformément un nombre $\leq x$, la probabilité qu'il soit premier est $1/\ln x$

Primalité

UNIVERSITÉ CAEN NORMANDIE

- Il existe une infinité de nombres premiers
- $\pi(x) = \#\{n \leq x : n \text{ est premier}\}$

  On a
  $$\pi(x) \sim x/\ln x$$
- Si on tire uniformément un nombre $\leq x$, la probabilité qu'il soit premier est $1/\ln x$

Deux problèmes :

- tester la primalité (Monté-Carlo)
- constuire des nombres premiers de taille fixée (Las Vegas)

Primalité

UNIVERSITÉ CAEN NORMANDIE

Théorème de Fermat

Soit p un entier premier et $a \in \mathbb{Z}$ tel que $\text{pgcd}(a, p) = 1$. Alors

$$a^{p-1} = 1 \mod p$$

- Test de primalité d'un entier N :

  Choisir $a \leq N$ aléatoirement et calculer $s = 2^{N-1} \mod N$

$$N \text{ premier} \Longrightarrow s = 1$$

$$s \neq 1 \Longrightarrow N \text{ non premier}$$

Primalité

UNIVERSITÉ CAEN NORMANDIE

Théorème de Fermat

Soit p un entier premier et $a \in \mathbb{Z}$ tel que $\text{pgcd}(a, p) = 1$. Alors

$$a^{p-1} = 1 \mod p$$

- Test de primalité d'un entier N :

  Choisir $a \leq N$ aléatoirement et calculer $s = 2^{N-1} \mod N$

$$N \text{ premier} \Longrightarrow s = 1$$
$$s \neq 1 \Longrightarrow N \text{ non premier}$$

Condition *nécessaire* de primalité.

Primalité

UNIVERSITÉ CAEN NORMANDIE

Théorème de Fermat

Soit p un entier premier et $a \in \mathbb{Z}$ tel que $\text{pgcd}(a, p) = 1$. Alors

$$a^{p-1} = 1 \mod p$$

- Test de primalité d'un entier N :

  Choisir $a \leq N$ aléatoirement et calculer $s = 2^{N-1} \mod N$

$$N \text{ premier} \Longrightarrow s = 1$$
$$s \neq 1 \Longrightarrow N \text{ non premier}$$

Condition *nécessaire* de primalité.
$N = 341,\ a = 2 \quad \rightsquigarrow \quad 2^{340} = 1 \mod 341$

Primalité

UNIVERSITÉ CAEN NORMANDIE

Théorème de Fermat

Soit p un entier premier et $a \in \mathbb{Z}$ tel que $\text{pgcd}(a, p) = 1$. Alors

$$a^{p-1} = 1 \mod p$$

- Test de primalité d'un entier N :

  Choisir $a \leq N$ aléatoirement et calculer $s = 2^{N-1} \mod N$

$$N \text{ premier} \Longrightarrow s = 1$$
$$s \neq 1 \Longrightarrow N \text{ non premier}$$

Condition *nécessaire* de primalité.
$N = 341,\ a = 2 \quad \rightsquigarrow \quad 2^{340} = 1 \mod 341 \quad$ mais $\quad 341 = 13 \times 11$

Primalité

UNIVERSITÉ CAEN NORMANDIE

Il existe des nombres entiers composés N, appelés *nombres de Carmichael* qui vérifient :

$$\forall \ a \in \mathbb{N}, \quad \text{pgcd}(a, N) = 1 \quad : \quad a^{N-1} = 1 \mod N$$

Autrement dit, ils passent toujours le précédent test de primalité ! ! !

$$\begin{aligned}
561 &= 3 \times 11 \times 17 \\
1105 &= 5 \times 13 \times 17 \\
1729 &= 7 \times 13 \times 19 \\
2465 &= 5 \times 17 \times 29 \\
2821 &= 7 \times 13 \times 31 \\
6601 &= 7 \times 23 \times 41 \\
8911 &= 7 \times 19 \times 67
\end{aligned}$$

Primalité

UNIVERSITÉ CAEN NORMANDIE

Théorème de Fermat

Soit p un entier premier et $a \in \mathbb{Z}$ tel que $\text{pgcd}(a, p) = 1$. Alors $a^{p-1} = 1 \mod p$

- Soit N un entier impair premier.

Primalité

UNIVERSITÉ CAEN NORMANDIE

Théorème de Fermat

Soit p un entier premier et $a \in \mathbb{Z}$ tel que $\text{pgcd}(a, p) = 1$. Alors $a^{p-1} = 1 \mod p$

- Soit N un entier impair premier.
- On peut écrire de manière unique $N - 1 = 2^s \cdot u$ avec u impair

Ex : $N = 101$ $N - 1 = 100 = 2^2 \cdot 25$

Primalité

UNIVERSITÉ CAEN NORMANDIE

Théorème de Fermat

Soit p un entier premier et $a \in \mathbb{Z}$ tel que $\text{pgcd}(a, p) = 1$. Alors $a^{p-1} = 1 \mod p$

- Soit N un entier impair premier.
- On peut écrire de manière unique $N - 1 = 2^s \cdot u$ avec u impair

$$\text{Ex :} \qquad N = 101 \qquad N - 1 = 100 = 2^2 \cdot 25$$

- En appliquant l'identité remarquable $(x-1)(x+1) = x^2 - 1$ plusieurs fois, nous avons

Primalité

UNIVERSITÉ CAEN NORMANDIE

Théorème de Fermat

Soit p un entier premier et $a \in \mathbb{Z}$ tel que $\text{pgcd}(a, p) = 1$. Alors $a^{p-1} = 1 \mod p$

- Soit N un entier impair premier.
- On peut écrire de manière unique $N - 1 = 2^s \cdot u$ avec u impair

$$\text{Ex :} \qquad N = 101 \qquad N - 1 = 100 = 2^2 \cdot 25$$

- En appliquant l'identité remarquable $(x-1)(x+1) = x^2 - 1$ plusieurs fois, nous avons

$$a^{N-1} - 1 = a^{2^s \cdot u} - 1 = (a^u - 1)(a^u + 1)(a^{2u} + 1)(a^{4u} + 1) \ldots (a^{2^{s-1}} + 1)$$

Primalité

UNIVERSITÉ CAEN NORMANDIE

Théorème de Fermat

Soit p un entier premier et $a \in \mathbb{Z}$ tel que $\text{pgcd}(a, p) = 1$. Alors $a^{p-1} = 1 \mod p$

- Soit N un entier impair premier.
- On peut écrire de manière unique $N - 1 = 2^s \cdot u$ avec u impair

  Ex : $N = 101$ $N - 1 = 100 = 2^2 \cdot 25$
- En appliquant l'identité remarquable $(x-1)(x+1) = x^2 - 1$ plusieurs fois, nous avons

  $a^{N-1} - 1 = a^{2^s \cdot u} - 1 = (a^u - 1)(a^u + 1)(a^{2u} + 1)(a^{4u} + 1) \ldots (a^{2^{s-1}} + 1)$

  Ex : $N = 101$ $a^{N-1} - 1 = a^{100} - 1 = (a^{25} - 1)(a^{25} + 1)(a^{50} - 1)$

Primalité

UNIVERSITÉ CAEN NORMANDIE

Théorème de Fermat

Soit p un entier premier et $a \in \mathbb{Z}$ tel que $\text{pgcd}(a, p) = 1$. Alors $a^{p-1} = 1 \mod p$

- Soit N un entier impair premier.
- On peut écrire de manière unique $N - 1 = 2^s \cdot u$ avec u impair

  Ex : $N = 101$ $N - 1 = 100 = 2^2 \cdot 25$
- En appliquant l'identité remarquable $(x-1)(x+1) = x^2 - 1$ plusieurs fois, nous avons

  $a^{N-1} - 1 = a^{2^s \cdot u} - 1 = (a^u - 1)(a^u + 1)(a^{2u} + 1)(a^{4u} + 1) \ldots (a^{2^{s-1}} + 1)$

  Ex : $N = 101$ $a^{N-1} - 1 = a^{100} - 1 = (a^{25} - 1)(a^{25} + 1)(a^{50} - 1)$
- Comme N est premier, le produit vaut $0 \mod N$ selon Fermat

Primalité

UNIVERSITÉ CAEN NORMANDIE

Théorème de Fermat

Soit p un entier premier et $a \in \mathbb{Z}$ tel que $\text{pgcd}(a, p) = 1$. Alors $a^{p-1} = 1 \mod p$

- Soit N un entier impair premier.
- On peut écrire de manière unique $N - 1 = 2^s \cdot u$ avec u impair

  Ex : $N = 101$ $N - 1 = 100 = 2^2 \cdot 25$
- En appliquant l'identité remarquable $(x-1)(x+1) = x^2 - 1$ plusieurs fois, nous avons

  $a^{N-1} - 1 = a^{2^s \cdot u} - 1 = (a^u - 1)(a^u + 1)(a^{2u} + 1)(a^{4u} + 1) \ldots (a^{2^{s-1}} + 1)$

  Ex : $N = 101$ $a^{N-1} - 1 = a^{100} - 1 = (a^{25} - 1)(a^{25} + 1)(a^{50} - 1)$
- Comme N est premier, le produit vaut $0 \mod N$ selon Fermat
- Autrement dit, l'une des parenthèses est nulle modulo N !!!

Primalité

UNIVERSITÉ CAEN NORMANDIE

Théorème de Fermat

Soit p un entier premier et $a \in \mathbb{Z}$ tel que $\text{pgcd}(a, p) = 1$. Alors $a^{p-1} = 1 \mod p$

- Soit N un entier impair premier.
- On peut écrire de manière unique $N - 1 = 2^s \cdot u$ avec u impair

  Ex : $N = 101$ $N - 1 = 100 = 2^2 \cdot 25$
- En appliquant l'identité remarquable $(x-1)(x+1) = x^2 - 1$ plusieurs fois, nous avons

  $a^{N-1} - 1 = a^{2^s \cdot u} - 1 = (a^u - 1)(a^u + 1)(a^{2u} + 1)(a^{4u} + 1) \ldots (a^{2^{s-1}} + 1)$

  Ex : $N = 101$ $a^{N-1} - 1 = a^{100} - 1 = (a^{25} - 1)(a^{25} + 1)(a^{50} - 1)$
- Comme N est premier, le produit vaut $0 \mod N$ selon Fermat
- Autrement dit, l'une des parenthèses est nulle modulo N !!!

  Ex : $N = 101,\ a = 2,\quad 2^{50} + 1 = 0 \mod 101$

Test de primalité de Miller-Rabin

UNIVERSITÉ CAEN NORMANDIE

Test de primalité de Miller-Rabin

Entrée : N un entier impair
Sortie : Retourne $non - premier$ ou $probablement - premier$
1 : Calculer s et u tels que $N - 1 = 2^s \cdot u$ avec u impair
2 : Choisir a premier avec N aléatoirement dans l'intervalle $]1, N[$
3 : **Vérifier si** $a^u - 1 \equiv 0 \mod N$ ou $a^u + 1 \equiv 0 \mod N$ ou
$a^{2u} + 1 \equiv 0 \mod N$ ou ... ou $a^{2^{s-1}u} + 1 \equiv 0 \mod N$
4 : **Si** l'une des égalités est vraie **Alors**
5 : **Retourner** $probablement - premier$
6 : **Sinon**
7 : **Retourner** $non - premier$
8 : **Fin Si**

Test de primalité de Miller-Rabin

UNIVERSITÉ CAEN NORMANDIE

Test de primalité de Miller-Rabin

Entrée : N un entier impair
Sortie : Retourne $non - premier$ ou $probablement - premier$
1 : Calculer s et u tels que $N - 1 = 2^s \cdot u$ avec u impair
2 : Choisir a premier avec N aléatoirement dans l'intervalle $]1, N[$
3 : **Vérifier si** $a^u - 1 \equiv 0 \mod N$ ou $a^u + 1 \equiv 0 \mod N$ ou
$a^{2u} + 1 \equiv 0 \mod N$ ou ... ou $a^{2^{s-1}u} + 1 \equiv 0 \mod N$
4 : **Si** l'une des égalités est vraie **Alors**
5 : **Retourner** $probablement - premier$
6 : **Sinon**
7 : **Retourner** $non - premier$
8 : **Fin Si**

- C'est un algorithme de Monté-Carlo biaisé vers le *faux* (non-premier)

Test de primalité de Miller-Rabin

UNIVERSITÉ CAEN NORMANDIE

Test de primalité de Miller-Rabin

Entrée : N un entier impair
Sortie : Retourne $non - premier$ ou $probablement - premier$
1 : Calculer s et u tels que $N - 1 = 2^s \cdot u$ avec u impair
2 : Choisir a premier avec N aléatoirement dans l'intervalle $]1, N[$
3 : **Vérifier si** $a^u - 1 \equiv 0 \mod N$ ou $a^u + 1 \equiv 0 \mod N$ ou
$a^{2u} + 1 \equiv 0 \mod N$ ou ... ou $a^{2^{s-1}u} + 1 \equiv 0 \mod N$
4 : **Si** l'une des égalités est vraie **Alors**
5 : **Retourner** $probablement - premier$
6 : **Sinon**
7 : **Retourner** $non - premier$
8 : **Fin Si**

- C'est un algorithme de Monté-Carlo biaisé vers le *faux* (non-premier)
- La probabilité d'erreur dans l'autre cas est $p_1 \leq \frac{1}{4}$

Test de primalité de Miller-Rabin

UNIVERSITÉ CAEN NORMANDIE

Test de primalité de Miller-Rabin

Entrée : N un entier impair
Sortie : Retourne $non - premier$ ou $probablement - premier$
1 : Calculer s et u tels que $N - 1 = 2^s \cdot u$ avec u impair
2 : Choisir a premier avec N aléatoirement dans l'intervalle $]1, N[$
3 : **Vérifier si** $a^u - 1 \equiv 0 \mod N$ ou $a^u + 1 \equiv 0 \mod N$ ou
$a^{2u} + 1 \equiv 0 \mod N$ ou ... ou $a^{2^{s-1}u} + 1 \equiv 0 \mod N$
4 : **Si** l'une des égalités est vraie **Alors**
5 : **Retourner** $probablement - premier$
6 : **Sinon**
7 : **Retourner** $non - premier$
8 : **Fin Si**

- C'est un algorithme de Monté-Carlo biaisé vers le *faux* (non-premier)
- La probabilité d'erreur dans l'autre cas est $p_1 \leq \frac{1}{4}$
- En effectuant au plus k appels au test, la probabilité d'erreur $p_k \leq 4^{-k}$

Algorithmes de type Las Vegas

UNIVERSITÉ CAEN NORMANDIE

Algorithme de type Las Vegas

Un algorithme de type Las Vegas est un algorithme probabiliste

- dont le résultat est toujours correct,
- mais dont l'espérance du temps de calcul est bornée :
  - pour toute entrée x, on pose $T(x)$ la variable aléatoire égale au temps de calcul sur x.
  - Alors il existe une fonction positive $f : \mathbb{N} \to \mathbb{R}_+$ telle que

$$\mathbb{E}[T(x)] \leq f(|x|)$$

avec $|x|$ la taille de l'entrée x (nombre de bits en mémoire)

Algorithmes de type Las Vegas

UNIVERSITÉ CAEN NORMANDIE

Exemple 1 :
Considérons un algorithme qui émet un 1 avec probabilité 1/3 et un 0 avec probabilité 2/3. L'algorithme réitère son expérience jusqu'à émettre un 1.

Quel est l'espérance du temps de calcul ?

Algorithmes de type Las Vegas

UNIVERSITÉ CAEN NORMANDIE

Exemple 1 :
Considérons un algorithme qui émet un 1 avec probabilité 1/3 et un 0 avec probabilité 2/3. L'algorithme réitère son expérience jusqu'à émettre un 1.

Quel est l'espérance du temps de calcul ?

$\rightarrow$ le nombre de tirages aléatoires suit loi géométrique de paramètre $p = 1/3$. L'espérance du nombre de tirages est donc $\frac{1}{p} = 3$.

Algorithmes de type Las Vegas

UNIVERSITÉ CAEN NORMANDIE

Exemple 1 :
Considérons un algorithme qui émet un 1 avec probabilité 1/3 et un 0 avec probabilité 2/3. L'algorithme réitère son expérience jusqu'à émettre un 1.

Quel est l'espérance du temps de calcul ?

$\rightarrow$ le nombre de tirages aléatoires suit loi géométrique de paramètre $p = 1/3$. L'espérance du nombre de tirages est donc $\frac{1}{p} = 3$.

Exemple 2 :
Génération de nombres premiers ....

Paradoxe des anniversaires

UNIVERSITÉ CAEN NORMANDIE

Question : Combien d'élèves faut-il dans une classe pour avoir plus d'une chance sur 2 que deux élèves soient nés le même jour ?

Réponse :

Paradoxe des anniversaires

UNIVERSITÉ CAEN NORMANDIE

Question : Combien d'élèves faut-il dans une classe pour avoir plus d'une chance sur 2 que deux élèves soient nés le même jour ?

Réponse :

- Le k-uplet $(x_1, \ldots, x_k)$ peut prendre 365^k valeurs possibles.

Paradoxe des anniversaires

UNIVERSITÉ CAEN NORMANDIE

Question : Combien d'élèves faut-il dans une classe pour avoir plus d'une chance sur 2 que deux élèves soient nés le même jour ?

Réponse :

- Le k-uplet $(x_1, \ldots, x_k)$ peut prendre 365^k valeurs possibles.
- Parmi ces valeurs, $365(365-1)(365-2)\cdots(365-k+1)$ ont des x_i tous différents.

Paradoxe des anniversaires

UNIVERSITÉ CAEN NORMANDIE

Question : Combien d'élèves faut-il dans une classe pour avoir plus d'une chance sur 2 que deux élèves soient nés le même jour ?

Réponse :

- Le k-uplet $(x_1, \ldots, x_k)$ peut prendre 365^k valeurs possibles.
- Parmi ces valeurs, $365(365-1)(365-2)\cdots(365-k+1)$ ont des x_i tous différents.
- La probabilité que k élèves sont nés des jours tous différents est

$$p_k = \frac{365(365-1)(365-2)\cdots(365-k+1)}{365^k} = \prod_{i=1}^{k-1}\left(1-\frac{i}{365}\right)$$

Paradoxe des anniversaires

UNIVERSITÉ CAEN NORMANDIE

Question : Combien d'élèves faut-il dans une classe pour avoir plus d'une chance sur 2 que deux élèves soient nés le même jour ?

Réponse :

- Le k-uplet $(x_1, \ldots, x_k)$ peut prendre 365^k valeurs possibles.
- Parmi ces valeurs, $365(365-1)(365-2)\cdots(365-k+1)$ ont des x_i tous différents.
- La probabilité que k élèves sont nés des jours tous différents est

$$p_k = \frac{365(365-1)(365-2)\cdots(365-k+1)}{365^k} = \prod_{i=1}^{k-1}\left(1-\frac{i}{365}\right)$$

- On a $p_{21} = 0.52$ et $p_{22} = 0.49$.

Paradoxe des anniversaires

UNIVERSITÉ CAEN NORMANDIE

Question : Combien d'élèves faut-il dans une classe pour avoir plus d'une chance sur 2 que deux élèves soient nés le même jour ?

Réponse :

- Le k-uplet $(x_1, \ldots, x_k)$ peut prendre 365^k valeurs possibles.
- Parmi ces valeurs, $365(365-1)(365-2)\cdots(365-k+1)$ ont des x_i tous différents.
- La probabilité que k élèves sont nés des jours tous différents est

$$p_k = \frac{365(365-1)(365-2)\cdots(365-k+1)}{365^k} = \prod_{i=1}^{k-1}\left(1-\frac{i}{365}\right)$$

- On a $p_{21} = 0.52$ et $p_{22} = 0.49$.
- A partir de 22 élèves, il y a plus d'une chance sur 2 que deux élèves soient nés le même jour !

Paradoxe des anniversaires : cas général

UNIVERSITÉ CAEN NORMANDIE

Question : Soit E un ensemble de cardinal n. Soit $(x_1, \ldots, x_k) \in E^k$ des éléments choisis aléatoirement et uniformément dans E. Pour quelle valeur de k a-t-on plus d'une chance sur 12 que deux x_i soient égaux ?

Réponse :

Paradoxe des anniversaires : cas général

UNIVERSITÉ CAEN NORMANDIE

Question : Soit E un ensemble de cardinal n. Soit $(x_1, \ldots, x_k) \in E^k$ des éléments choisis aléatoirement et uniformément dans E. Pour quelle valeur de k a-t-on plus d'une chance sur 12 que deux x_i soient égaux ?

Réponse :

- Le k-uplet $(x_1, \ldots, x_k)$ peut prendre n^k valeurs possibles.

Paradoxe des anniversaires : cas général

UNIVERSITÉ CAEN NORMANDIE

Question : Soit E un ensemble de cardinal n. Soit $(x_1, \ldots, x_k) \in E^k$ des éléments choisis aléatoirement et uniformément dans E. Pour quelle valeur de k a-t-on plus d'une chance sur 12 que deux x_i soient égaux ?

Réponse :

- Le k-uplet $(x_1, \ldots, x_k)$ peut prendre n^k valeurs possibles.
- Parmi ces valeurs, seules $n(n-1)(n-2)\cdots(n-k+1)$ ont des x_i différents 2 à 2.

Paradoxe des anniversaires : cas général

UNIVERSITÉ CAEN NORMANDIE

Question : Soit E un ensemble de cardinal n. Soit $(x_1, \ldots, x_k) \in E^k$ des éléments choisis aléatoirement et uniformément dans E. Pour quelle valeur de k a-t-on plus d'une chance sur 12 que deux x_i soient égaux ?

Réponse :

- Le k-uplet $(x_1, \ldots, x_k)$ peut prendre n^k valeurs possibles.
- Parmi ces valeurs, seules $n(n-1)(n-2)\cdots(n-k+1)$ ont des x_i différents 2 à 2.
- La probabilité que les k x_i sont différents est

$$p_k = \frac{n(n-1)(n-2)\cdots(n-k+1)}{n^k} = \prod_{i=1}^{k-1}\left(1-\frac{i}{n}\right)$$

$$p_k \leq \exp\left(\frac{-1}{n}\sum_{i=1}^{k-1} i\right) = \exp\left(\frac{-k(k-1)}{2n}\right).$$

Paradoxe des anniversaires : cas général

UNIVERSITÉ CAEN NORMANDIE

Question : Soit E un ensemble de cardinal n. Soit $(x_1, \ldots, x_k) \in E^k$ des éléments choisis aléatoirement et uniformément dans E. Pour quelle valeur de k a-t-on plus d'une chance sur 12 que deux x_i soient égaux ?

Réponse :

- Le k-uplet $(x_1, \ldots, x_k)$ peut prendre n^k valeurs possibles.
- Parmi ces valeurs, seules $n(n-1)(n-2)\cdots(n-k+1)$ ont des x_i différents 2 à 2.
- La probabilité que les k x_i sont différents est

$$p_k = \frac{n(n-1)(n-2)\cdots(n-k+1)}{n^k} = \prod_{i=1}^{k-1}\left(1-\frac{i}{n}\right)$$

$$p_k \leq \exp\left(\frac{-1}{n}\sum_{i=1}^{k-1} i\right) = \exp\left(\frac{-k(k-1)}{2n}\right).$$

- si $k \sim \sqrt{2n\ln 12}$ alor $p_k \approx 1/12$

Paradoxe des anniversaires : cas général

UNIVERSITÉ CAEN NORMANDIE

Question : Soit E un ensemble de cardinal n. Soit $(x_1, \ldots, x_k) \in E^k$ des éléments choisis aléatoirement et uniformément dans E. Pour quelle valeur de k a-t-on plus d'une chance sur 12 que deux x_i soient égaux ?

Réponse :

- Le k-uplet $(x_1, \ldots, x_k)$ peut prendre n^k valeurs possibles.
- Parmi ces valeurs, seules $n(n-1)(n-2)\cdots(n-k+1)$ ont des x_i différents 2 à 2.
- La probabilité que les k x_i sont différents est p_k
- si $k \sim \sqrt{2n \ln 12}$ alors $p_k \approx 1/12$

Conclusion : Il suffit que k soit de l'ordre de $\sqrt{n}$ pour que la probabilité de ne pas avoir une collision soit plus petite que $1/2$.

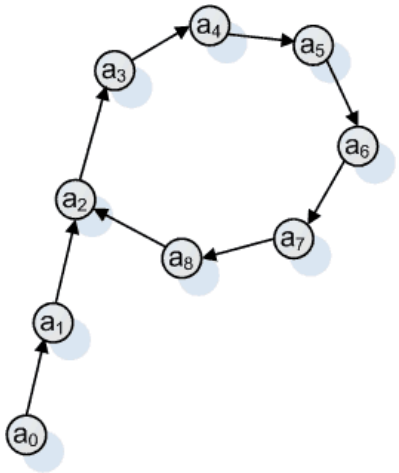
Présentation du problème

Soit E un ensemble de cardinal n.
Soit $f : E \to E$ une fonction.
Soit a_0 un élément de E. On pose

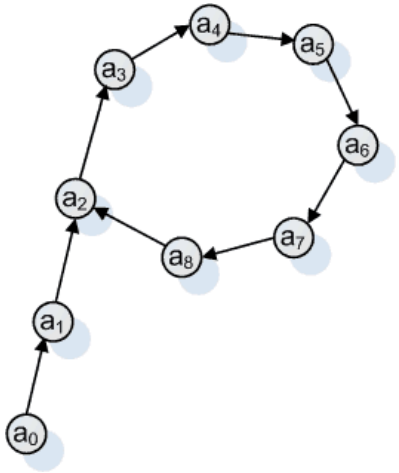
$$a_{i+1} = f(a_i).$$

Problème

La suite $(a_i)_{i \in \mathbb{N}}$ est ultimement périodique de période ℓ_c. Déterminer la valeur ℓ_c de la période (aussi appelée cycle).

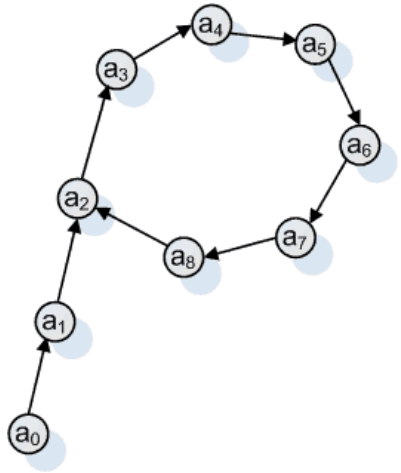


Présentation du problème



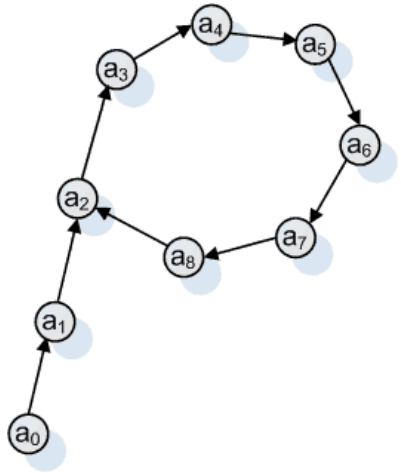
- Exemple : $f(x) = x^2 + 1 \mod 20$

Présentation du problème



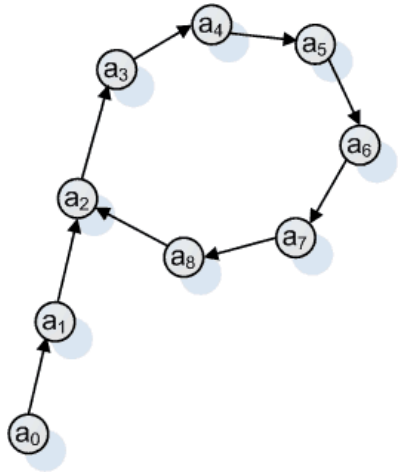
- Exemple : $f(x) = x^2 + 1 \mod 20$
  3,

Présentation du problème



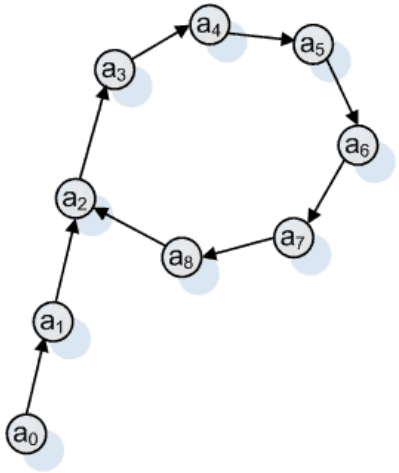
- Exemple : $f(x) = x^2 + 1 \mod 20$
  3, 10,

Présentation du problème



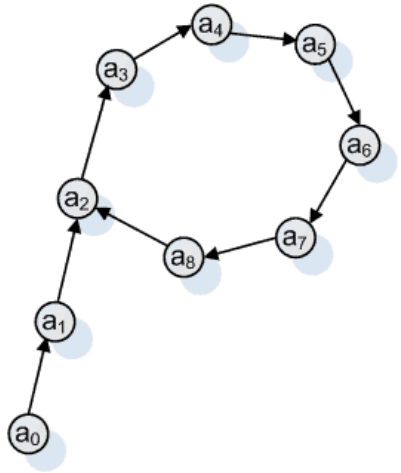
- Exemple : $f(x) = x^2 + 1 \mod 20$
  3, 10, 1,

Présentation du problème



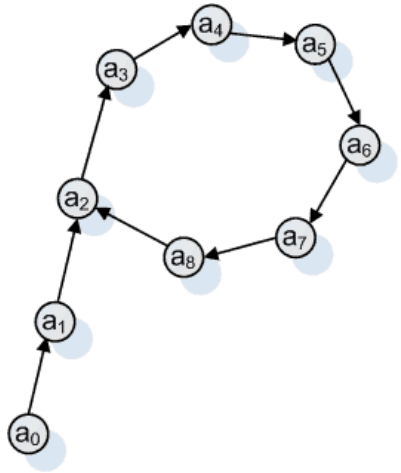
- Exemple : $f(x) = x^2 + 1 \mod 20$
  3, 10, 1, 2,

Présentation du problème



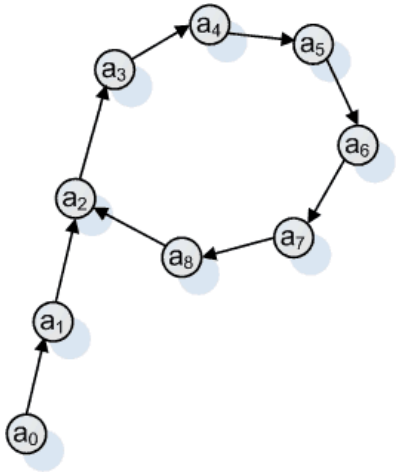
- Exemple : $f(x) = x^2 + 1 \mod 20$
  3, 10, 1, 2, 5,

Présentation du problème



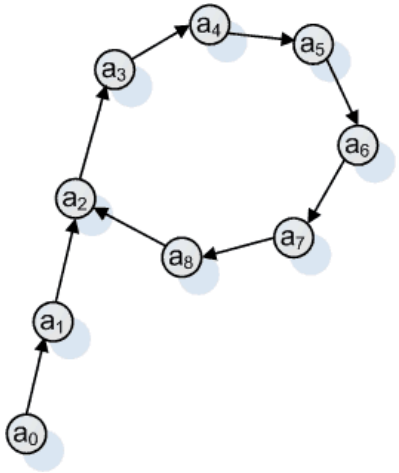
- Exemple : $f(x) = x^2 + 1 \mod 20$
  3, 10, 1, 2, 5, 6,

Présentation du problème



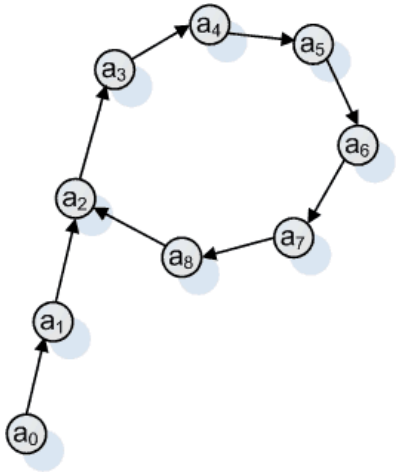
- Exemple : $f(x) = x^2 + 1 \mod 20$
  3, 10, 1, 2, 5, 6, 20,

Présentation du problème



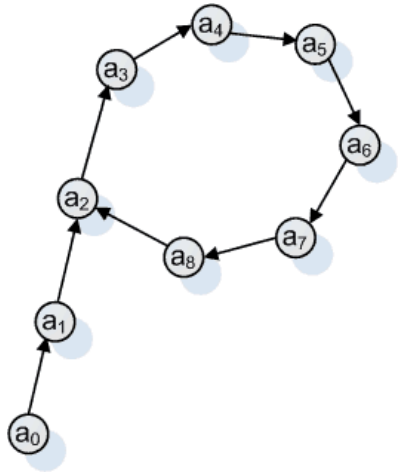
- Exemple : $f(x) = x^2 + 1 \mod 20$
  3, 10, 1, 2, 5, 6, 20, 17,

Présentation du problème



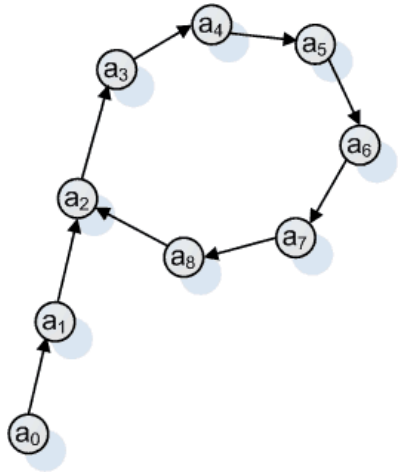
- Exemple : $f(x) = x^2 + 1 \mod 20$
  3, 10, 1, 2, 5, 6, 20, 17, 10,

Présentation du problème



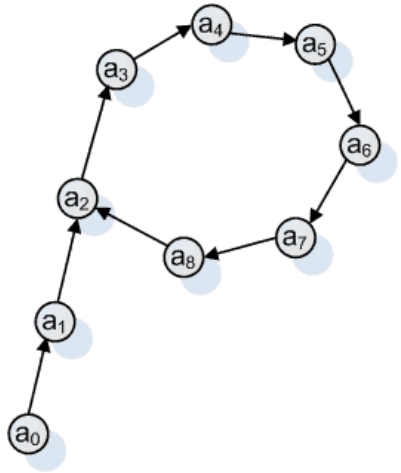
- Exemple : $f(x) = x^2 + 1 \mod 20$
  3, 10, 1, 2, 5, 6, 20, 17, 10, 1, 2, ...
- Exemple : $f(x) = x^2 + 1 \mod 521$

Présentation du problème



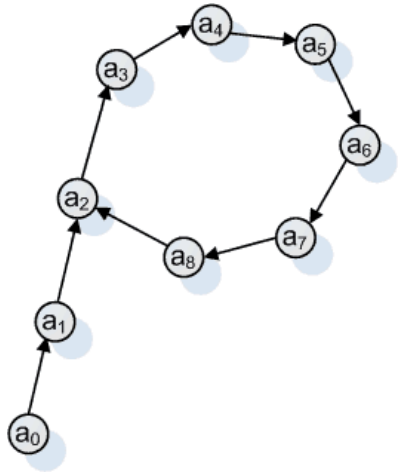
- Exemple : $f(x) = x^2 + 1 \mod 20$
  3, 10, 1, 2, 5, 6, 20, 17, 10, 1, 2, ...
- Exemple : $f(x) = x^2 + 1 \mod 521$
  32,

Présentation du problème



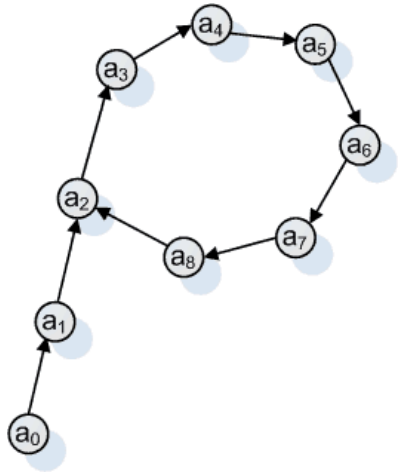
- Exemple : $f(x) = x^2 + 1 \mod 20$
  3, 10, 1, 2, 5, 6, 20, 17, 10, 1, 2, ...
- Exemple : $f(x) = x^2 + 1 \mod 521$
  32, 504,

Présentation du problème



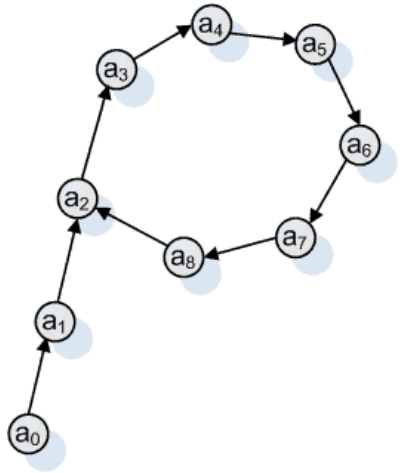
- Exemple : $f(x) = x^2 + 1 \mod 20$
  3, 10, 1, 2, 5, 6, 20, 17, 10, 1, 2, ...
- Exemple : $f(x) = x^2 + 1 \mod 521$
  32, 504, 290,

Présentation du problème



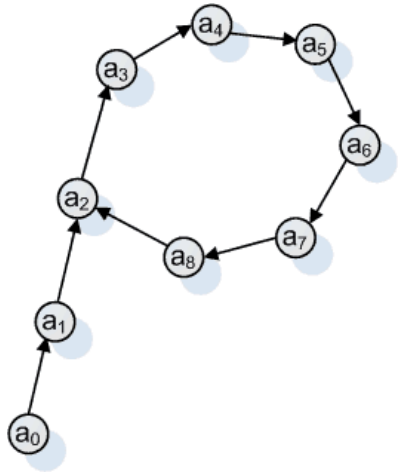
- Exemple : $f(x) = x^2 + 1 \mod 20$
  3, 10, 1, 2, 5, 6, 20, 17, 10, 1, 2, ...
- Exemple : $f(x) = x^2 + 1 \mod 521$
  32, 504, 290, 220,

Présentation du problème



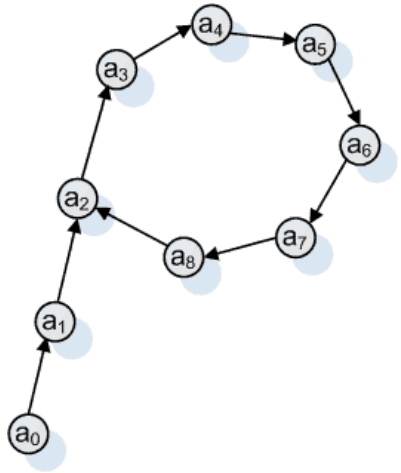
- Exemple : $f(x) = x^2 + 1 \mod 20$
  3, 10, 1, 2, 5, 6, 20, 17, 10, 1, 2, ...
- Exemple : $f(x) = x^2 + 1 \mod 521$
  32, 504, 290, 220, 469,

Présentation du problème



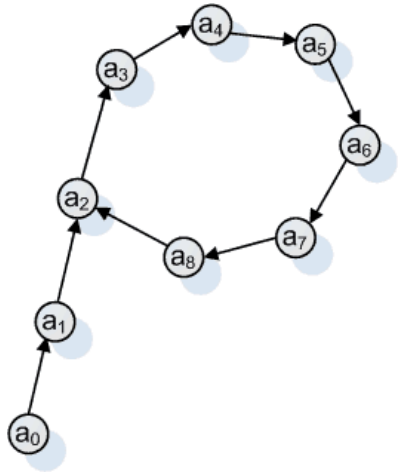
- Exemple : $f(x) = x^2 + 1 \mod 20$
  3, 10, 1, 2, 5, 6, 20, 17, 10, 1, 2, ...
- Exemple : $f(x) = x^2 + 1 \mod 521$
  32, 504, 290, 220, 469, 100

Présentation du problème



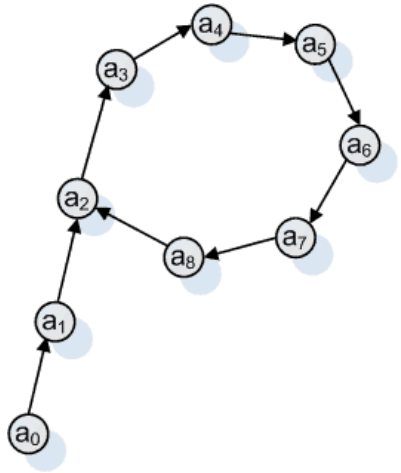
- Exemple : $f(x) = x^2 + 1 \mod 20$
  3, 10, 1, 2, 5, 6, 20, 17, 10, 1, 2, ...
- Exemple : $f(x) = x^2 + 1 \mod 521$
  32, 504, 290, 220, 469, 100, 102

Présentation du problème



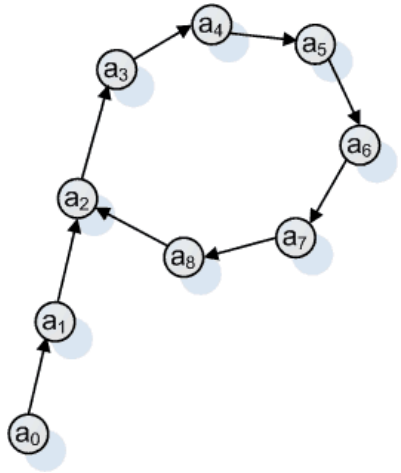
- Exemple : $f(x) = x^2 + 1 \mod 20$
  3, 10, 1, 2, 5, 6, 20, 17, 10, 1, 2, ...
- Exemple : $f(x) = x^2 + 1 \mod 521$
  32, 504, 290, 220, 469, 100, 102, 506

Présentation du problème



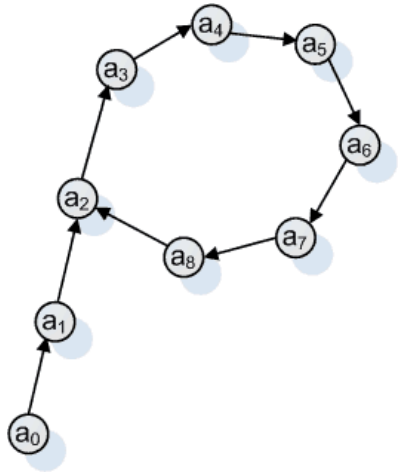
- Exemple : $f(x) = x^2 + 1 \mod 20$
  3, 10, 1, 2, 5, 6, 20, 17, 10, 1, 2, ...
- Exemple : $f(x) = x^2 + 1 \mod 521$
  32, 504, 290, 220, 469, 100, 102, 506, 226

Présentation du problème



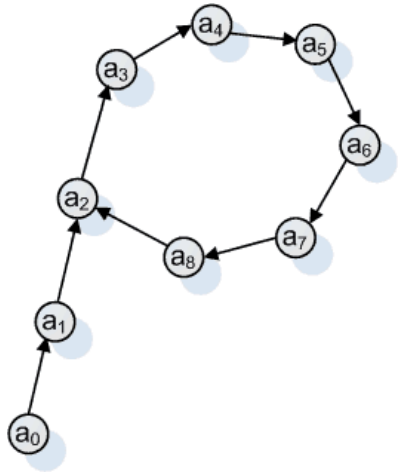
- Exemple : $f(x) = x^2 + 1 \mod 20$
  3, 10, 1, 2, 5, 6, 20, 17, 10, 1, 2, ...
- Exemple : $f(x) = x^2 + 1 \mod 521$
  32, 504, 290, 220, 469, 100, 102, 506, 226, 19

Présentation du problème



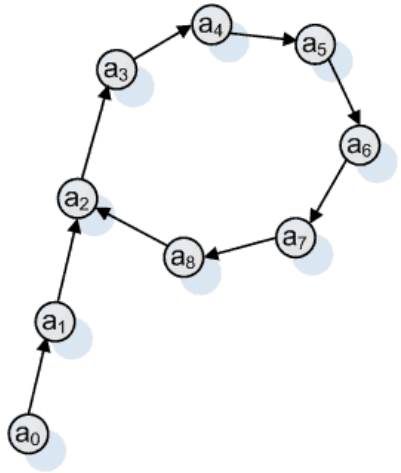
- Exemple : $f(x) = x^2 + 1 \mod 20$
  3, 10, 1, 2, 5, 6, 20, 17, 10, 1, 2, ...
- Exemple : $f(x) = x^2 + 1 \mod 521$
  32, 504, 290, 220, 469, 100, 102, 506, 226, 19, 362

Présentation du problème



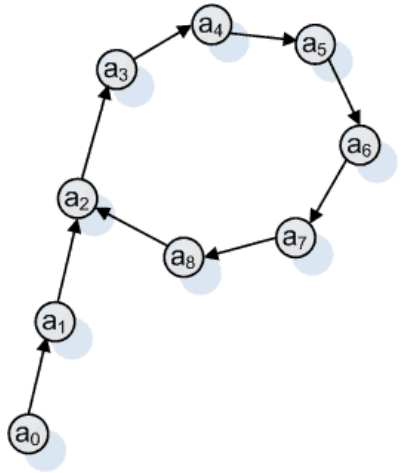
- Exemple : $f(x) = x^2 + 1 \mod 20$
  3, 10, 1, 2, 5, 6, 20, 17, 10, 1, 2, ...
- Exemple : $f(x) = x^2 + 1 \mod 521$
  32, 504, 290, 220, 469, 100, 102, 506, 226, 19, 362 , 274, 53, 205, 346, 408, 266, 422, 424, 32, 504, ...

Présentation du problème



- Exemple : $f(x) = x^2 + 1 \mod 20$
  3, 10, 1, 2, 5, 6, 20, 17, 10, 1, 2, ...
- Exemple : $f(x) = x^2 + 1 \mod 521$
  32, 504, 290, 220, 469, 100, 102, 506, 226, 19, 362 , 274, 53, 205, 346, 408, 266, 422, 424, 32, 504, ...
- Remarque : entre les deux 32 $\rightsquigarrow$ 18 éléments

Présentation du problème



- Exemple : $f(x) = x^2 + 1 \mod 20$
  3, 10, 1, 2, 5, 6, 20, 17, 10, 1, 2, ...
- Exemple : $f(x) = x^2 + 1 \mod 521$
  32, 504, 290, 220, 469, 100, 102, 506, 226, 19, 362 , 274, 53, 205, 346, 408, 266, 422, 424, 32, 504, ...
- Remarque : entre les deux 32 $\rightsquigarrow$ 18 éléments
  $\sqrt{521} \simeq 22,8$.

Algorithme naïf

Entrée : la fonction $f : E \to E$ et $a_0 \in E$
Sortie : La longueur du cycle de la suite $(a_i)_{i \in \mathbb{N}}$

$S = \{\}$
$a = a_0$
$i = 0$
Stocker $(a, 0)$ dans l'ensemble S
Tant que true faire
    $a = f(a)$
    $i = i + 1$
    Si il existe (a, j) dans S **alors**
        retourner $i - j$
    sinon
        Ajouter (a, i) à S.
    fin si
fin Tant que

Algorithme naïf

Entrée : la fonction $f : E \to E$ et $a_0 \in E$
Sortie : La longueur du cycle de la suite $(a_i)_{i \in \mathbb{N}}$

$S = \{\}$
$a = a_0$
$i = 0$
Stocker $(a, 0)$ dans l'ensemble S
Tant que true faire
  $a = f(a)$
  $i = i + 1$
  Si il existe (a, j) dans S **alors**
    retourner $i - j$
  sinon
    Ajouter (a, i) à S.
  fin si
fin Tant que

Complexité :

- temps : $O(\mathrm{card}(E))$ dans le pire des cas

Algorithme naïf

Entrée : la fonction $f : E \to E$ et $a_0 \in E$
Sortie : La longueur du cycle de la suite $(a_i)_{i \in \mathbb{N}}$

$S = \{\}$
$a = a_0$
$i = 0$
Stocker $(a, 0)$ dans l'ensemble S
Tant que true faire
  $a = f(a)$
  $i = i + 1$
  Si il existe (a, j) dans S **alors**
    retourner $i - j$
  sinon
    Ajouter (a, i) à S.
  fin si
fin Tant que

Complexité :

- temps : $O(\mathrm{card}(E))$ dans le pire des cas
- mémoire : $O(\mathrm{card}(E))$ dans le pire des cas

Algorithme naïf

Entrée : la fonction $f : E \to E$ et $a_0 \in E$
Sortie : La longueur du cycle de la suite $(a_i)_{i \in \mathbb{N}}$

$S = \{\}$
$a = a_0$
$i = 0$
Stocker $(a, 0)$ dans l'ensemble S
Tant que true faire
  $a = f(a)$
  $i = i + 1$
  Si il existe (a, j) dans S **alors**
    retourner $i - j$
  sinon
    Ajouter (a, i) à S.
  fin si
fin Tant que

Complexité :

- temps : $O(\mathrm{card}(E))$ dans le pire des cas
- mémoire : $O(\mathrm{card}(E))$ dans le pire des cas
- temps moyen : $O(\sqrt{\mathrm{card}(E)})$ (paradoxe des anniversaires)

Algorithme naïf

Entrée : la fonction $f : E \to E$ et $a_0 \in E$
Sortie : La longueur du cycle de la suite $(a_i)_{i \in \mathbb{N}}$

$S = \{\}$
$a = a_0$
$i = 0$
Stocker $(a, 0)$ dans l'ensemble S
Tant que true faire
  $a = f(a)$
  $i = i + 1$
  Si il existe (a, j) dans S **alors**
    retourner $i - j$
  sinon
    Ajouter (a, i) à S.
  fin si
fin Tant que

Complexité :

- temps : $O(\mathrm{card}(E))$ dans le pire des cas
- mémoire : $O(\mathrm{card}(E))$ dans le pire des cas
- temps moyen : $O(\sqrt{\mathrm{card}(E)})$ (paradoxe des anniversaires)
- espace moyen : $O(\sqrt{\mathrm{card}(E)})$

Exemple d'exécution

On pose $E = \{0, 1, 2, \ldots, 100\}$, $f(x) = x^2 + 1$, $a_0 = 3$

Algorithme naïf :

$a_0 = 3$

Exemple d'exécution

On pose $E = \{0, 1, 2, \ldots, 100\}$, $f(x) = x^2 + 1$, $a_0 = 3$

Algorithme naïf :

$a_0 = 3$ $a_1 = 10$

Exemple d'exécution

UNIVERSITÉ CAEN NORMANDIE

On pose $E = \{0, 1, 2, \ldots, 100\}$, $f(x) = x^2 + 1$, $a_0 = 3$

Algorithme naïf :

$a_0 = 3$ $a_1 = 10$ $a_2 = 0$

Exemple d'exécution

On pose $E = \{0, 1, 2, \ldots, 100\}$, $f(x) = x^2 + 1$, $a_0 = 3$

Algorithme naïf :

$a_0 = 3$ $a_1 = 10$ $a_2 = 0$ $a_3 = 1$

Exemple d'exécution

On pose $E = \{0, 1, 2, \ldots, 100\}$, $f(x) = x^2 + 1$, $a_0 = 3$

Algorithme naïf :

$a_0 = 3$ $a_1 = 10$ $a_2 = 0$ $a_3 = 1$ $a_4 = 2$

Exemple d'exécution

On pose $E = \{0, 1, 2, \ldots, 100\}$, $f(x) = x^2 + 1$, $a_0 = 3$

Algorithme naïf :

$$a_0 = 3 \qquad a_1 = 10 \qquad a_2 = 0 \qquad a_3 = 1 \qquad a_4 = 2 \qquad a_5 = 5$$

Exemple d'exécution

On pose $E = \{0, 1, 2, \ldots, 100\}$, $f(x) = x^2 + 1$, $a_0 = 3$

Algorithme naïf :

$a_0 = 3$ $a_1 = 10$ $a_2 = 0$ $a_3 = 1$ $a_4 = 2$ $a_5 = 5$
$a_6 = 26$

Exemple d'exécution

On pose $E = \{0, 1, 2, \ldots, 100\}$, $f(x) = x^2 + 1$, $a_0 = 3$

Algorithme naïf :

$a_0 = 3$ $a_1 = 10$ $a_2 = 0$ $a_3 = 1$ $a_4 = 2$ $a_5 = 5$
$a_6 = 26$ $a_7 = 71$

Exemple d'exécution

On pose $E = \{0, 1, 2, \ldots, 100\}$, $f(x) = x^2 + 1$, $a_0 = 3$

Algorithme naïf :

$a_0 = 3$ $a_1 = 10$ $a_2 = 0$ $a_3 = 1$ $a_4 = 2$ $a_5 = 5$
$a_6 = 26$ $a_7 = 71$ $a_8 = 93$

Exemple d'exécution

On pose $E = \{0, 1, 2, \ldots, 100\}$, $f(x) = x^2 + 1$, $a_0 = 3$

Algorithme naïf :

$a_0 = 3$ $a_1 = 10$ $a_2 = 0$ $a_3 = 1$ $a_4 = 2$ $a_5 = 5$
$a_6 = 26$ $a_7 = 71$ $a_8 = 93$ $a_9 = 65$

Exemple d'exécution

On pose $E = \{0, 1, 2, \ldots, 100\}$, $f(x) = x^2 + 1$, $a_0 = 3$

Algorithme naïf :

$$a_0 = 3 \quad a_1 = 10 \quad a_2 = 0 \quad a_3 = 1 \quad a_4 = 2 \quad a_5 = 5$$
$$a_6 = 26 \quad a_7 = 71 \quad a_8 = 93 \quad a_9 = 65 \quad a_{10} = 85$$

Exemple d'exécution

On pose $E = \{0, 1, 2, \ldots, 100\}$, $f(x) = x^2 + 1$, $a_0 = 3$

Algorithme naïf :

$$a_0 = 3 \quad a_1 = 10 \quad a_2 = 0 \quad a_3 = 1 \quad a_4 = 2 \quad a_5 = 5$$
$$a_6 = 26 \quad a_7 = 71 \quad a_8 = 93 \quad a_9 = 65 \quad a_{10} = 85 \quad a_{11} = 55$$

Exemple d'exécution

On pose $E = \{0, 1, 2, \ldots, 100\}$, $f(x) = x^2 + 1$, $a_0 = 3$

Algorithme naïf :

$$a_0 = 3 \quad a_1 = 10 \quad a_2 = 0 \quad a_3 = 1 \quad a_4 = 2 \quad a_5 = 5$$
$$a_6 = 26 \quad a_7 = 71 \quad a_8 = 93 \quad a_9 = 65 \quad a_{10} = 85 \quad a_{11} = 55$$
$$a_{12} = 97$$

Exemple d'exécution

On pose $E = \{0, 1, 2, \ldots, 100\}$, $f(x) = x^2 + 1$, $a_0 = 3$

Algorithme naïf :

$$a_0 = 3 \quad a_1 = 10 \quad a_2 = 0 \quad a_3 = 1 \quad a_4 = 2 \quad a_5 = 5$$
$$a_6 = 26 \quad a_7 = 71 \quad a_8 = 93 \quad a_9 = 65 \quad a_{10} = 85 \quad a_{11} = 55$$
$$a_{12} = 97 \quad a_{13} = 17$$

Exemple d'exécution

On pose $E = \{0, 1, 2, \ldots, 100\}$, $f(x) = x^2 + 1$, $a_0 = 3$

Algorithme naïf :

$$a_0 = 3 \quad a_1 = 10 \quad a_2 = 0 \quad a_3 = 1 \quad a_4 = 2 \quad a_5 = 5$$
$$a_6 = 26 \quad a_7 = 71 \quad a_8 = 93 \quad a_9 = 65 \quad a_{10} = 85 \quad a_{11} = 55$$
$$a_{12} = 97 \quad a_{13} = 17 \quad a_{14} = 88$$

Exemple d'exécution

On pose $E = \{0, 1, 2, \ldots, 100\}$, $f(x) = x^2 + 1$, $a_0 = 3$

Algorithme naïf :

$$a_0 = 3 \quad a_1 = 10 \quad a_2 = 0 \quad a_3 = 1 \quad a_4 = 2 \quad a_5 = 5$$
$$a_6 = 26 \quad a_7 = 71 \quad a_8 = 93 \quad a_9 = 65 \quad a_{10} = 85 \quad a_{11} = 55$$
$$a_{12} = 97 \quad a_{13} = 17 \quad a_{14} = 88 \quad a_{15} = 69$$

Exemple d'exécution

On pose $E = \{0, 1, 2, \ldots, 100\}$, $f(x) = x^2 + 1$, $a_0 = 3$

Algorithme naïf :

$$
\begin{array}{cccccc}
a_0 = 3 & a_1 = 10 & a_2 = 0 & a_3 = 1 & a_4 = 2 & a_5 = 5 \\
a_6 = 26 & a_7 = 71 & a_8 = 93 & a_9 = 65 & a_{10} = 85 & a_{11} = 55 \\
a_{12} = 97 & a_{13} = 17 & a_{14} = 88 & a_{15} = 69 & a_{16} = 15 &
\end{array}
$$

Exemple d'exécution

On pose $E = \{0, 1, 2, \ldots, 100\}$, $f(x) = x^2 + 1$, $a_0 = 3$

Algorithme naïf :

$$
\begin{array}{cccccc}
a_0 = 3 & a_1 = 10 & a_2 = 0 & a_3 = 1 & a_4 = 2 & a_5 = 5 \\
a_6 = 26 & a_7 = 71 & a_8 = 93 & a_9 = 65 & a_{10} = 85 & a_{11} = 55 \\
a_{12} = 97 & a_{13} = 17 & a_{14} = 88 & a_{15} = 69 & a_{16} = 15 & a_{17} = 24
\end{array}
$$

Exemple d'exécution

On pose $E = \{0, 1, 2, \ldots, 100\}$, $f(x) = x^2 + 1$, $a_0 = 3$

Algorithme naïf :

$$a_0 = 3 \quad a_1 = 10 \quad a_2 = 0 \quad a_3 = 1 \quad a_4 = 2 \quad a_5 = 5$$
$$a_6 = 26 \quad a_7 = 71 \quad a_8 = 93 \quad a_9 = 65 \quad a_{10} = 85 \quad a_{11} = 55$$
$$a_{12} = 97 \quad a_{13} = 17 \quad a_{14} = 88 \quad a_{15} = 69 \quad a_{16} = 15 \quad a_{17} = 24$$
$$a_{18} = 72$$

Exemple d'exécution

On pose $E = \{0, 1, 2, \ldots, 100\}$, $f(x) = x^2 + 1$, $a_0 = 3$

Algorithme naïf :

$$
\begin{array}{cccccc}
a_0 = 3 & a_1 = 10 & a_2 = 0 & a_3 = 1 & a_4 = 2 & a_5 = 5 \\
a_6 = 26 & a_7 = 71 & a_8 = 93 & a_9 = 65 & a_{10} = 85 & a_{11} = 55 \\
a_{12} = 97 & a_{13} = 17 & a_{14} = 88 & a_{15} = 69 & a_{16} = 15 & a_{17} = 24 \\
a_{18} = 72 & a_{19} = 34 & & & &
\end{array}
$$

Exemple d'exécution

UNIVERSITÉ CAEN NORMANDIE

On pose $E = \{0, 1, 2, \ldots, 100\}$, $f(x) = x^2 + 1$, $a_0 = 3$

Algorithme naïf :

$a_0 = 3$	$a_1 = 10$	$a_2 = 0$	$a_3 = 1$	$a_4 = 2$	$a_5 = 5$
$a_6 = 26$	$a_7 = 71$	$a_8 = 93$	$a_9 = 65$	$a_{10} = 85$	$a_{11} = 55$
$a_{12} = 97$	$a_{13} = 17$	$a_{14} = 88$	$a_{15} = 69$	$a_{16} = 15$	$a_{17} = 24$
$a_{18} = 72$	$a_{19} = 34$	$a_{20} = 46$			

Exemple d'exécution

On pose $E = \{0, 1, 2, \ldots, 100\}$, $f(x) = x^2 + 1$, $a_0 = 3$

Algorithme naïf :

$a_0 = 3$	$a_1 = 10$	$a_2 = 0$	$a_3 = 1$	$a_4 = 2$	$a_5 = 5$
$a_6 = 26$	$a_7 = 71$	$a_8 = 93$	$a_9 = 65$	$a_{10} = 85$	$a_{11} = 55$
$a_{12} = 97$	$a_{13} = 17$	$a_{14} = 88$	$a_{15} = 69$	$a_{16} = 15$	$a_{17} = 24$
$a_{18} = 72$	$a_{19} = 34$	$a_{20} = 46$	$a_{21} = 97$		

Algorithme de détection de cycle de Floyd

Entrée : la fonction $f : E \to E$ et $a_0 \in E$

Sortie : La longueur μ du cycle de la suite $(a_i)_{i \in \mathbb{N}}$

$m = 0; \quad x = y = a_0$

Faire

$m = m + 1$

$x = f(x) \qquad (x \leftarrow a_m)$

$y = f(f(y)) \quad (y \leftarrow a_{2m})$

Tant que $x \neq y$

$m_1 = m$

Faire

$m_1 = m_1 + 1$

$x = f(x) \qquad (x \leftarrow a_{m_1})$

$y = f(f(y)) \quad (y \leftarrow a_{2m_1})$

Tant que $x \neq y$

Retourner $\mu = m_1 - m$

Algorithme de détection de cycle de Floyd

Entrée : la fonction $f : E \to E$ et $a_0 \in E$
Sortie : La longueur μ du cycle de la suite $(a_i)_{i \in \mathbb{N}}$

$m = 0; \quad x = y = a_0$
Faire
  $m = m + 1$
  $x = f(x) \qquad (x \leftarrow a_m)$
  $y = f(f(y)) \quad (y \leftarrow a_{2m})$
Tant que $x \neq y$
$m_1 = m$
Faire
  $m_1 = m_1 + 1$
  $x = f(x) \qquad (x \leftarrow a_{m_1})$
  $y = f(f(y)) \quad (y \leftarrow a_{2m_1})$
Tant que $x \neq y$
Retourner $\mu = m_1 - m$

Principes :

- à la sortie de la première boucle, $a_m = a_{2m}$

Algorithme de détection de cycle de Floyd

Entrée : la fonction $f : E \to E$ et $a_0 \in E$
Sortie : La longueur μ du cycle de la suite $(a_i)_{i\in\mathbb{N}}$

$m = 0; \quad x = y = a_0$
Faire
  $m = m + 1$
  $x = f(x) \qquad (x \leftarrow a_m)$
  $y = f(f(y)) \quad (y \leftarrow a_{2m})$
Tant que $x \neq y$
$m_1 = m$
Faire
  $m_1 = m_1 + 1$
  $x = f(x) \qquad (x \leftarrow a_{m_1})$
  $y = f(f(y)) \quad (y \leftarrow a_{2m_1})$
Tant que $x \neq y$
Retourner $\mu = m_1 - m$

Principes :

- à la sortie de la première boucle, $a_m = a_{2m}$
- soit $2m - m = m$ est un multiple de la longueur du cycle

Algorithme de détection de cycle de Floyd

Entrée : la fonction $f : E \to E$ et $a_0 \in E$
Sortie : La longueur μ du cycle de la suite $(a_i)_{i\in\mathbb{N}}$

$m = 0; \quad x = y = a_0$
Faire
  $m = m + 1$
  $x = f(x) \qquad (x \leftarrow a_m)$
  $y = f(f(y)) \quad (y \leftarrow a_{2m})$
Tant que $x \neq y$
$m_1 = m$
Faire
  $m_1 = m_1 + 1$
  $x = f(x) \qquad (x \leftarrow a_{m_1})$
  $y = f(f(y)) \quad (y \leftarrow a_{2m_1})$
Tant que $x \neq y$
Retourner $\mu = m_1 - m$

Principes :

- à la sortie de la première boucle, $a_m = a_{2m}$
- soit $2m - m = m$ est un multiple de la longueur du cycle
- après la deuxième boucle, $a_{m_1} = a_{2m_1}$ et $m_1 \leq m + \mu$

Algorithme de détection de cycle de Floyd

Entrée : la fonction $f : E \to E$ et $a_0 \in E$
Sortie : La longueur μ du cycle de la suite $(a_i)_{i\in\mathbb{N}}$

$m = 0$; $\quad x = y = a_0$
Faire
  $m = m + 1$
  $x = f(x) \qquad (x \leftarrow a_m)$
  $y = f(f(y)) \quad (y \leftarrow a_{2m})$
Tant que $x \neq y$
$m_1 = m$
Faire
  $m_1 = m_1 + 1$
  $x = f(x) \qquad (x \leftarrow a_{m_1})$
  $y = f(f(y)) \quad (y \leftarrow a_{2m_1})$
Tant que $x \neq y$
Retourner $\mu = m_1 - m$

Principes :

- à la sortie de la première boucle, $a_m = a_{2m}$
- soit $2m - m = m$ est un multiple de la longueur du cycle
- après la deuxième boucle, $a_{m_1} = a_{2m_1}$ et $m_1 \leq m + \mu$
- $2m_1 - m_1 = m_1$ est aussi un multiple de μ soit $\mu = m_1 - m$.

Algorithme de détection de cycle de Floyd

Entrée : la fonction $f : E \to E$ et $a_0 \in E$
Sortie : La longueur μ du cycle de la suite $(a_i)_{i\in\mathbb{N}}$

$m = 0$; $\quad x = y = a_0$
Faire
  $m = m + 1$
  $x = f(x) \qquad (x \leftarrow a_m)$
  $y = f(f(y)) \quad (y \leftarrow a_{2m})$
Tant que $x \neq y$
$m_1 = m$
Faire
  $m_1 = m_1 + 1$
  $x = f(x) \qquad (x \leftarrow a_{m_1})$
  $y = f(f(y)) \quad (y \leftarrow a_{2m_1})$
Tant que $x \neq y$
Retourner $\mu = m_1 - m$

Principes :

- à la sortie de la première boucle, $a_m = a_{2m}$
- soit $2m - m = m$ est un multiple de la longueur du cycle
- après la deuxième boucle, $a_{m_1} = a_{2m_1}$ et $m_1 \leq m + \mu$
- $2m_1 - m_1 = m_1$ est aussi un multiple de μ soit $\mu = m_1 - m$.
- temps moyen : $O(\sqrt{\mathrm{card}(E)})$ (paradoxe des anniversaires)

Algorithme de détection de cycle de Floyd

Entrée : la fonction $f : E \to E$ et $a_0 \in E$
Sortie : La longueur μ du cycle de la suite $(a_i)_{i\in\mathbb{N}}$

$m = 0$; $\quad x = y = a_0$
Faire
  $m = m + 1$
  $x = f(x) \qquad (x \leftarrow a_m)$
  $y = f(f(y)) \quad (y \leftarrow a_{2m})$
Tant que $x \neq y$
$m_1 = m$
Faire
  $m_1 = m_1 + 1$
  $x = f(x) \qquad (x \leftarrow a_{m_1})$
  $y = f(f(y)) \quad (y \leftarrow a_{2m_1})$
Tant que $x \neq y$
Retourner $\mu = m_1 - m$

Principes :

- à la sortie de la première boucle, $a_m = a_{2m}$
- soit $2m - m = m$ est un multiple de la longueur du cycle
- après la deuxième boucle, $a_{m_1} = a_{2m_1}$ et $m_1 \leq m + \mu$
- $2m_1 - m_1 = m_1$ est aussi un multiple de μ soit $\mu = m_1 - m$.
- temps moyen : $O(\sqrt{\mathrm{card}(E)})$ (paradoxe des anniversaires)
- espace moyen $O(1)$

Exemple d'exécution

On pose $E = \{0, 1, 2, \ldots, 100\}$, $f(x) = x^2 + 1 \mod 100$, $a_0 = 3$

Algorithme de Floyd :

Phase 1 :

$$\begin{cases} a_0 = 3 \\ a_0 = 3 \end{cases}$$

Exemple d'exécution

On pose $E = \{0, 1, 2, \ldots, 100\}$, $f(x) = x^2 + 1 \mod 100$, $a_0 = 3$

Algorithme de Floyd :

Phase 1 :

$$\begin{cases} a_0 = 3 \\ a_0 = 3 \end{cases} \quad \begin{cases} a_1 = 10 \\ a_2 = 0 \end{cases}$$

Exemple d'exécution

On pose $E = \{0, 1, 2, \ldots, 100\}$, $f(x) = x^2 + 1 \mod 100$, $a_0 = 3$

Algorithme de Floyd :

Phase 1 :

$$\begin{cases} a_0 = 3 \\ a_0 = 3 \end{cases} \quad \begin{cases} a_1 = 10 \\ a_2 = 0 \end{cases} \quad \begin{cases} a_2 = 0 \\ a_4 = 2 \end{cases}$$

Exemple d'exécution

On pose $E = \{0, 1, 2, \ldots, 100\}$, $f(x) = x^2 + 1 \mod 100$, $a_0 = 3$

Algorithme de Floyd :

Phase 1 :

$$\left\{ \begin{array}{l} a_0 = 3 \\ a_0 = 3 \end{array} \right. \quad \left\{ \begin{array}{l} a_1 = 10 \\ a_2 = 0 \end{array} \right. \quad \left\{ \begin{array}{l} a_2 = 0 \\ a_4 = 2 \end{array} \right. \quad \left\{ \begin{array}{l} a_3 = 1 \\ a_6 = 26 \end{array} \right.$$

Exemple d'exécution

On pose $E = \{0, 1, 2, \ldots, 100\}$, $f(x) = x^2 + 1 \mod 100$, $a_0 = 3$

Algorithme de Floyd :

Phase 1 :

$$\left\{ \begin{array}{l} a_0 = 3 \\ a_0 = 3 \end{array} \right. \quad \left\{ \begin{array}{l} a_1 = 10 \\ a_2 = 0 \end{array} \right. \quad \left\{ \begin{array}{l} a_2 = 0 \\ a_4 = 2 \end{array} \right. \quad \left\{ \begin{array}{l} a_3 = 1 \\ a_6 = 26 \end{array} \right. \quad \left\{ \begin{array}{l} a_4 = 2 \\ a_8 = 93 \end{array} \right.$$

Exemple d'exécution

On pose $E = \{0, 1, 2, \ldots, 100\}$, $f(x) = x^2 + 1 \mod 100$, $a_0 = 3$

Algorithme de Floyd :

Phase 1 :

$$\left\{ \begin{array}{l} a_0 = 3 \\ a_0 = 3 \end{array} \right. \quad \left\{ \begin{array}{l} a_1 = 10 \\ a_2 = 0 \end{array} \right. \quad \left\{ \begin{array}{l} a_2 = 0 \\ a_4 = 2 \end{array} \right. \quad \left\{ \begin{array}{l} a_3 = 1 \\ a_6 = 26 \end{array} \right. \quad \left\{ \begin{array}{l} a_4 = 2 \\ a_8 = 93 \end{array} \right. \quad \left\{ \begin{array}{l} a_5 = 5 \\ a_{10} = 85 \end{array} \right.$$

Exemple d'exécution

On pose $E = \{0, 1, 2, \ldots, 100\}$, $f(x) = x^2 + 1 \mod 100$, $a_0 = 3$

Algorithme de Floyd :

Phase 1 :

$$\left\{ \begin{array}{l} a_0 = 3 \\ a_0 = 3 \end{array} \right. \quad \left\{ \begin{array}{l} a_1 = 10 \\ a_2 = 0 \end{array} \right. \quad \left\{ \begin{array}{l} a_2 = 0 \\ a_4 = 2 \end{array} \right. \quad \left\{ \begin{array}{l} a_3 = 1 \\ a_6 = 26 \end{array} \right. \quad \left\{ \begin{array}{l} a_4 = 2 \\ a_8 = 93 \end{array} \right. \quad \left\{ \begin{array}{l} a_5 = 5 \\ a_{10} = 85 \end{array} \right.$$

$$\left\{ \begin{array}{l} a_6 = 26 \\ a_{12} = 97 \end{array} \right.$$

Exemple d'exécution

On pose $E = \{0, 1, 2, \ldots, 100\}$, $f(x) = x^2 + 1 \mod 100$, $a_0 = 3$

Algorithme de Floyd :

Phase 1 :

$$\begin{cases} a_0 = 3 \\ a_0 = 3 \end{cases} \quad \begin{cases} a_1 = 10 \\ a_2 = 0 \end{cases} \quad \begin{cases} a_2 = 0 \\ a_4 = 2 \end{cases} \quad \begin{cases} a_3 = 1 \\ a_6 = 26 \end{cases} \quad \begin{cases} a_4 = 2 \\ a_8 = 93 \end{cases} \quad \begin{cases} a_5 = 5 \\ a_{10} = 85 \end{cases}$$

$$\begin{cases} a_6 = 26 \\ a_{12} = 97 \end{cases} \quad \begin{cases} a_7 = 71 \\ a_{14} = 88 \end{cases}$$

Exemple d'exécution

On pose $E = \{0, 1, 2, \ldots, 100\}$, $f(x) = x^2 + 1 \mod 100$, $a_0 = 3$

Algorithme de Floyd :

Phase 1 :

$$\left\{ \begin{array}{l} a_0 = 3 \\ a_0 = 3 \end{array} \right. \quad \left\{ \begin{array}{l} a_1 = 10 \\ a_2 = 0 \end{array} \right. \quad \left\{ \begin{array}{l} a_2 = 0 \\ a_4 = 2 \end{array} \right. \quad \left\{ \begin{array}{l} a_3 = 1 \\ a_6 = 26 \end{array} \right. \quad \left\{ \begin{array}{l} a_4 = 2 \\ a_8 = 93 \end{array} \right. \quad \left\{ \begin{array}{l} a_5 = 5 \\ a_{10} = 85 \end{array} \right.$$

$$\left\{ \begin{array}{l} a_6 = 26 \\ a_{12} = 97 \end{array} \right. \quad \left\{ \begin{array}{l} a_7 = 71 \\ a_{14} = 88 \end{array} \right. \quad \left\{ \begin{array}{l} a_8 = 93 \\ a_{16} = 15 \end{array} \right.$$

Exemple d'exécution

On pose $E = \{0, 1, 2, \ldots, 100\}$, $f(x) = x^2 + 1 \mod 100$, $a_0 = 3$

Algorithme de Floyd :

Phase 1 :

$$\left\{\begin{array}{l} a_0 = 3 \\ a_0 = 3 \end{array}\right. \quad \left\{\begin{array}{l} a_1 = 10 \\ a_2 = 0 \end{array}\right. \quad \left\{\begin{array}{l} a_2 = 0 \\ a_4 = 2 \end{array}\right. \quad \left\{\begin{array}{l} a_3 = 1 \\ a_6 = 26 \end{array}\right. \quad \left\{\begin{array}{l} a_4 = 2 \\ a_8 = 93 \end{array}\right. \quad \left\{\begin{array}{l} a_5 = 5 \\ a_{10} = 85 \end{array}\right.$$

$$\left\{\begin{array}{l} a_6 = 26 \\ a_{12} = 97 \end{array}\right. \quad \left\{\begin{array}{l} a_7 = 71 \\ a_{14} = 88 \end{array}\right. \quad \left\{\begin{array}{l} a_8 = 93 \\ a_{16} = 15 \end{array}\right. \quad \left\{\begin{array}{l} a_9 = 65 \\ a_{18} = 72 \end{array}\right.$$

Exemple d'exécution

On pose $E = \{0, 1, 2, \ldots, 100\}$, $f(x) = x^2 + 1 \mod 100$, $a_0 = 3$

Algorithme de Floyd :

Phase 1 :

$$\left\{ \begin{array}{l} a_0 = 3 \\ a_0 = 3 \end{array} \right. \quad \left\{ \begin{array}{l} a_1 = 10 \\ a_2 = 0 \end{array} \right. \quad \left\{ \begin{array}{l} a_2 = 0 \\ a_4 = 2 \end{array} \right. \quad \left\{ \begin{array}{l} a_3 = 1 \\ a_6 = 26 \end{array} \right. \quad \left\{ \begin{array}{l} a_4 = 2 \\ a_8 = 93 \end{array} \right. \quad \left\{ \begin{array}{l} a_5 = 5 \\ a_{10} = 85 \end{array} \right.$$

$$\left\{ \begin{array}{l} a_6 = 26 \\ a_{12} = 97 \end{array} \right. \quad \left\{ \begin{array}{l} a_7 = 71 \\ a_{14} = 88 \end{array} \right. \quad \left\{ \begin{array}{l} a_8 = 93 \\ a_{16} = 15 \end{array} \right. \quad \left\{ \begin{array}{l} a_9 = 65 \\ a_{18} = 72 \end{array} \right. \quad \left\{ \begin{array}{l} a_{10} = 85 \\ a_{20} = 46 \end{array} \right.$$

Exemple d'exécution

On pose $E = \{0, 1, 2, \ldots, 100\}$, $f(x) = x^2 + 1 \mod 100$, $a_0 = 3$

Algorithme de Floyd :

Phase 1 :

$$\left\{\begin{array}{l} a_0 = 3 \\ a_0 = 3 \end{array}\right. \quad \left\{\begin{array}{l} a_1 = 10 \\ a_2 = 0 \end{array}\right. \quad \left\{\begin{array}{l} a_2 = 0 \\ a_4 = 2 \end{array}\right. \quad \left\{\begin{array}{l} a_3 = 1 \\ a_6 = 26 \end{array}\right. \quad \left\{\begin{array}{l} a_4 = 2 \\ a_8 = 93 \end{array}\right. \quad \left\{\begin{array}{l} a_5 = 5 \\ a_{10} = 85 \end{array}\right.$$

$$\left\{\begin{array}{l} a_6 = 26 \\ a_{12} = 97 \end{array}\right. \quad \left\{\begin{array}{l} a_7 = 71 \\ a_{14} = 88 \end{array}\right. \quad \left\{\begin{array}{l} a_8 = 93 \\ a_{16} = 15 \end{array}\right. \quad \left\{\begin{array}{l} a_9 = 65 \\ a_{18} = 72 \end{array}\right. \quad \left\{\begin{array}{l} a_{10} = 85 \\ a_{20} = 46 \end{array}\right. \quad \left\{\begin{array}{l} a_{11} = 55 \\ a_{22} = 17 \end{array}\right.$$

Exemple d'exécution

On pose $E = \{0, 1, 2, \ldots, 100\}$, $f(x) = x^2 + 1 \mod 100$, $a_0 = 3$

Algorithme de Floyd :

Phase 1 :

$$\begin{cases} a_0 = 3 \\ a_0 = 3 \end{cases} \quad \begin{cases} a_1 = 10 \\ a_2 = 0 \end{cases} \quad \begin{cases} a_2 = 0 \\ a_4 = 2 \end{cases} \quad \begin{cases} a_3 = 1 \\ a_6 = 26 \end{cases} \quad \begin{cases} a_4 = 2 \\ a_8 = 93 \end{cases} \quad \begin{cases} a_5 = 5 \\ a_{10} = 85 \end{cases}$$

$$\begin{cases} a_6 = 26 \\ a_{12} = 97 \end{cases} \quad \begin{cases} a_7 = 71 \\ a_{14} = 88 \end{cases} \quad \begin{cases} a_8 = 93 \\ a_{16} = 15 \end{cases} \quad \begin{cases} a_9 = 65 \\ a_{18} = 72 \end{cases} \quad \begin{cases} a_{10} = 85 \\ a_{20} = 46 \end{cases} \quad \begin{cases} a_{11} = 55 \\ a_{22} = 17 \end{cases}$$

$$\begin{cases} a_{12} = 97 \\ a_{24} = 69 \end{cases}$$

Exemple d'exécution

On pose $E = \{0, 1, 2, \ldots, 100\}$, $f(x) = x^2 + 1 \mod 100$, $a_0 = 3$

Algorithme de Floyd :

Phase 1 :

$$\begin{cases} a_0 = 3 \\ a_0 = 3 \end{cases} \quad \begin{cases} a_1 = 10 \\ a_2 = 0 \end{cases} \quad \begin{cases} a_2 = 0 \\ a_4 = 2 \end{cases} \quad \begin{cases} a_3 = 1 \\ a_6 = 26 \end{cases} \quad \begin{cases} a_4 = 2 \\ a_8 = 93 \end{cases} \quad \begin{cases} a_5 = 5 \\ a_{10} = 85 \end{cases}$$

$$\begin{cases} a_6 = 26 \\ a_{12} = 97 \end{cases} \quad \begin{cases} a_7 = 71 \\ a_{14} = 88 \end{cases} \quad \begin{cases} a_8 = 93 \\ a_{16} = 15 \end{cases} \quad \begin{cases} a_9 = 65 \\ a_{18} = 72 \end{cases} \quad \begin{cases} a_{10} = 85 \\ a_{20} = 46 \end{cases} \quad \begin{cases} a_{11} = 55 \\ a_{22} = 17 \end{cases}$$

$$\begin{cases} a_{12} = 97 \\ a_{24} = 69 \end{cases} \quad \begin{cases} a_{13} = 17 \\ a_{26} = 24 \end{cases}$$

Exemple d'exécution

On pose $E = \{0, 1, 2, \ldots, 100\}$, $f(x) = x^2 + 1 \mod 100$, $a_0 = 3$

Algorithme de Floyd :

Phase 1 :

$$\begin{cases} a_0 = 3 \\ a_0 = 3 \end{cases} \quad \begin{cases} a_1 = 10 \\ a_2 = 0 \end{cases} \quad \begin{cases} a_2 = 0 \\ a_4 = 2 \end{cases} \quad \begin{cases} a_3 = 1 \\ a_6 = 26 \end{cases} \quad \begin{cases} a_4 = 2 \\ a_8 = 93 \end{cases} \quad \begin{cases} a_5 = 5 \\ a_{10} = 85 \end{cases}$$

$$\begin{cases} a_6 = 26 \\ a_{12} = 97 \end{cases} \quad \begin{cases} a_7 = 71 \\ a_{14} = 88 \end{cases} \quad \begin{cases} a_8 = 93 \\ a_{16} = 15 \end{cases} \quad \begin{cases} a_9 = 65 \\ a_{18} = 72 \end{cases} \quad \begin{cases} a_{10} = 85 \\ a_{20} = 46 \end{cases} \quad \begin{cases} a_{11} = 55 \\ a_{22} = 17 \end{cases}$$

$$\begin{cases} a_{12} = 97 \\ a_{24} = 69 \end{cases} \quad \begin{cases} a_{13} = 17 \\ a_{26} = 24 \end{cases} \quad \begin{cases} a_{14} = 88 \\ a_{28} = 34 \end{cases}$$

Exemple d'exécution

On pose $E = \{0, 1, 2, \ldots, 100\}$, $f(x) = x^2 + 1 \mod 100$, $a_0 = 3$

Algorithme de Floyd :

Phase 1 :

$$\begin{cases} a_0 = 3 \\ a_0 = 3 \end{cases} \quad \begin{cases} a_1 = 10 \\ a_2 = 0 \end{cases} \quad \begin{cases} a_2 = 0 \\ a_4 = 2 \end{cases} \quad \begin{cases} a_3 = 1 \\ a_6 = 26 \end{cases} \quad \begin{cases} a_4 = 2 \\ a_8 = 93 \end{cases} \quad \begin{cases} a_5 = 5 \\ a_{10} = 85 \end{cases}$$

$$\begin{cases} a_6 = 26 \\ a_{12} = 97 \end{cases} \quad \begin{cases} a_7 = 71 \\ a_{14} = 88 \end{cases} \quad \begin{cases} a_8 = 93 \\ a_{16} = 15 \end{cases} \quad \begin{cases} a_9 = 65 \\ a_{18} = 72 \end{cases} \quad \begin{cases} a_{10} = 85 \\ a_{20} = 46 \end{cases} \quad \begin{cases} a_{11} = 55 \\ a_{22} = 17 \end{cases}$$

$$\begin{cases} a_{12} = 97 \\ a_{24} = 69 \end{cases} \quad \begin{cases} a_{13} = 17 \\ a_{26} = 24 \end{cases} \quad \begin{cases} a_{14} = 88 \\ a_{28} = 34 \end{cases} \quad \begin{cases} a_{15} = 69 \\ a_{30} = 97 \end{cases}$$

Exemple d'exécution

On pose $E = \{0, 1, 2, \ldots, 100\}$, $f(x) = x^2 + 1 \mod 100$, $a_0 = 3$

Algorithme de Floyd :

Phase 1 :

$$\begin{cases} a_0 = 3 \\ a_0 = 3 \end{cases} \quad \begin{cases} a_1 = 10 \\ a_2 = 0 \end{cases} \quad \begin{cases} a_2 = 0 \\ a_4 = 2 \end{cases} \quad \begin{cases} a_3 = 1 \\ a_6 = 26 \end{cases} \quad \begin{cases} a_4 = 2 \\ a_8 = 93 \end{cases} \quad \begin{cases} a_5 = 5 \\ a_{10} = 85 \end{cases}$$

$$\begin{cases} a_6 = 26 \\ a_{12} = 97 \end{cases} \quad \begin{cases} a_7 = 71 \\ a_{14} = 88 \end{cases} \quad \begin{cases} a_8 = 93 \\ a_{16} = 15 \end{cases} \quad \begin{cases} a_9 = 65 \\ a_{18} = 72 \end{cases} \quad \begin{cases} a_{10} = 85 \\ a_{20} = 46 \end{cases} \quad \begin{cases} a_{11} = 55 \\ a_{22} = 17 \end{cases}$$

$$\begin{cases} a_{12} = 97 \\ a_{24} = 69 \end{cases} \quad \begin{cases} a_{13} = 17 \\ a_{26} = 24 \end{cases} \quad \begin{cases} a_{14} = 88 \\ a_{28} = 34 \end{cases} \quad \begin{cases} a_{15} = 69 \\ a_{30} = 97 \end{cases} \quad \begin{cases} a_{16} = 15 \\ a_{32} = 88 \end{cases}$$

Exemple d'exécution

On pose $E = \{0, 1, 2, \ldots, 100\}$, $f(x) = x^2 + 1 \mod 100$, $a_0 = 3$

Algorithme de Floyd :

Phase 1 :

$$\begin{cases} a_0 = 3 \\ a_0 = 3 \end{cases} \quad \begin{cases} a_1 = 10 \\ a_2 = 0 \end{cases} \quad \begin{cases} a_2 = 0 \\ a_4 = 2 \end{cases} \quad \begin{cases} a_3 = 1 \\ a_6 = 26 \end{cases} \quad \begin{cases} a_4 = 2 \\ a_8 = 93 \end{cases} \quad \begin{cases} a_5 = 5 \\ a_{10} = 85 \end{cases}$$

$$\begin{cases} a_6 = 26 \\ a_{12} = 97 \end{cases} \quad \begin{cases} a_7 = 71 \\ a_{14} = 88 \end{cases} \quad \begin{cases} a_8 = 93 \\ a_{16} = 15 \end{cases} \quad \begin{cases} a_9 = 65 \\ a_{18} = 72 \end{cases} \quad \begin{cases} a_{10} = 85 \\ a_{20} = 46 \end{cases} \quad \begin{cases} a_{11} = 55 \\ a_{22} = 17 \end{cases}$$

$$\begin{cases} a_{12} = 97 \\ a_{24} = 69 \end{cases} \quad \begin{cases} a_{13} = 17 \\ a_{26} = 24 \end{cases} \quad \begin{cases} a_{14} = 88 \\ a_{28} = 34 \end{cases} \quad \begin{cases} a_{15} = 69 \\ a_{30} = 97 \end{cases} \quad \begin{cases} a_{16} = 15 \\ a_{32} = 88 \end{cases} \quad \begin{cases} a_{17} = 24 \\ a_{34} = 15 \end{cases}$$

Exemple d'exécution

On pose $E = \{0, 1, 2, \ldots, 100\}$, $f(x) = x^2 + 1 \mod 100$, $a_0 = 3$

Algorithme de Floyd :

Phase 1 :

$$\begin{cases} a_0 = 3 \\ a_0 = 3 \end{cases} \quad \begin{cases} a_1 = 10 \\ a_2 = 0 \end{cases} \quad \begin{cases} a_2 = 0 \\ a_4 = 2 \end{cases} \quad \begin{cases} a_3 = 1 \\ a_6 = 26 \end{cases} \quad \begin{cases} a_4 = 2 \\ a_8 = 93 \end{cases} \quad \begin{cases} a_5 = 5 \\ a_{10} = 85 \end{cases}$$

$$\begin{cases} a_6 = 26 \\ a_{12} = 97 \end{cases} \quad \begin{cases} a_7 = 71 \\ a_{14} = 88 \end{cases} \quad \begin{cases} a_8 = 93 \\ a_{16} = 15 \end{cases} \quad \begin{cases} a_9 = 65 \\ a_{18} = 72 \end{cases} \quad \begin{cases} a_{10} = 85 \\ a_{20} = 46 \end{cases} \quad \begin{cases} a_{11} = 55 \\ a_{22} = 17 \end{cases}$$

$$\begin{cases} a_{12} = 97 \\ a_{24} = 69 \end{cases} \quad \begin{cases} a_{13} = 17 \\ a_{26} = 24 \end{cases} \quad \begin{cases} a_{14} = 88 \\ a_{28} = 34 \end{cases} \quad \begin{cases} a_{15} = 69 \\ a_{30} = 97 \end{cases} \quad \begin{cases} a_{16} = 15 \\ a_{32} = 88 \end{cases} \quad \begin{cases} a_{17} = 24 \\ a_{34} = 15 \end{cases}$$

$$\begin{cases} a_{18} = 72 \\ a_{36} = 72 \end{cases} \quad \Rightarrow m = 18$$

Exemple d'exécution

On pose $E = \{0, 1, 2, \ldots, 100\}$, $f(x) = x^2 + 1 \mod 100$, $a_0 = 3$

Algorithme de Floyd :

Phase 1 :

$$\begin{cases} a_{18} = 72 \\ a_{36} = 72 \end{cases} \Rightarrow m = 18$$

Phase 2 :

Exemple d'exécution

On pose $E = \{0, 1, 2, \ldots, 100\}$, $f(x) = x^2 + 1 \mod 100$, $a_0 = 3$

Algorithme de Floyd :

Phase 1 :

$$\begin{cases} a_{18} = 72 \\ a_{36} = 72 \end{cases} \Rightarrow m = 18$$

Phase 2 :

$$\begin{cases} a_{19} = 34 \\ a_{38} = 46 \end{cases}$$

Exemple d'exécution

On pose $E = \{0, 1, 2, \ldots, 100\}$, $f(x) = x^2 + 1 \mod 100$, $a_0 = 3$

Algorithme de Floyd :

Phase 1 :

$$\begin{cases} a_{18} = 72 \\ a_{36} = 72 \end{cases} \Rightarrow m = 18$$

Phase 2 :

$$\begin{cases} a_{19} = 34 \\ a_{38} = 46 \end{cases} \quad \begin{cases} a_{20} = 46 \\ a_{40} = 17 \end{cases}$$

Exemple d'exécution

On pose $E = \{0, 1, 2, \ldots, 100\}$, $f(x) = x^2 + 1 \mod 100$, $a_0 = 3$

Algorithme de Floyd :

Phase 1 :

$$\begin{cases} a_{18} = 72 \\ a_{36} = 72 \end{cases} \Rightarrow m = 18$$

Phase 2 :

$$\begin{cases} a_{19} = 34 \\ a_{38} = 46 \end{cases} \quad \begin{cases} a_{20} = 46 \\ a_{40} = 17 \end{cases} \quad \begin{cases} a_{21} = 97 \\ a_{42} = 69 \end{cases}$$

Exemple d'exécution

On pose $E = \{0, 1, 2, \ldots, 100\}$, $f(x) = x^2 + 1 \mod 100$, $a_0 = 3$

Algorithme de Floyd :

Phase 1 :

$$\begin{cases} a_{18} = 72 \\ a_{36} = 72 \end{cases} \Rightarrow m = 18$$

Phase 2 :

$$\begin{cases} a_{19} = 34 \\ a_{38} = 46 \end{cases} \quad \begin{cases} a_{20} = 46 \\ a_{40} = 17 \end{cases} \quad \begin{cases} a_{21} = 97 \\ a_{42} = 69 \end{cases} \quad \begin{cases} a_{22} = 17 \\ a_{44} = 24 \end{cases}$$

Exemple d'exécution

UNIVERSITÉ CAEN NORMANDIE

On pose $E = \{0, 1, 2, \ldots, 100\}$, $f(x) = x^2 + 1 \mod 100$, $a_0 = 3$

Algorithme de Floyd :

Phase 1 :

$$\begin{cases} a_{18} = 72 \\ a_{36} = 72 \end{cases} \Rightarrow m = 18$$

Phase 2 :

$$\begin{cases} a_{19} = 34 \\ a_{38} = 46 \end{cases} \quad \begin{cases} a_{20} = 46 \\ a_{40} = 17 \end{cases} \quad \begin{cases} a_{21} = 97 \\ a_{42} = 69 \end{cases} \quad \begin{cases} a_{22} = 17 \\ a_{44} = 24 \end{cases} \quad \begin{cases} a_{23} = 88 \\ a_{46} = 34 \end{cases}$$

Exemple d'exécution

On pose $E = \{0, 1, 2, \ldots, 100\}$, $f(x) = x^2 + 1 \mod 100$, $a_0 = 3$

Algorithme de Floyd :

Phase 1 :

$$\begin{cases} a_{18} = 72 \\ a_{36} = 72 \end{cases} \Rightarrow m = 18$$

Phase 2 :

$$\begin{cases} a_{19} = 34 \\ a_{38} = 46 \end{cases} \begin{cases} a_{20} = 46 \\ a_{40} = 17 \end{cases} \begin{cases} a_{21} = 97 \\ a_{42} = 69 \end{cases} \begin{cases} a_{22} = 17 \\ a_{44} = 24 \end{cases} \begin{cases} a_{23} = 88 \\ a_{46} = 34 \end{cases} \begin{cases} a_{24} = 69 \\ a_{48} = 97 \end{cases}$$

Exemple d'exécution

On pose $E = \{0, 1, 2, \ldots, 100\}$, $f(x) = x^2 + 1 \mod 100$, $a_0 = 3$

Algorithme de Floyd :

Phase 1 :

$$\begin{cases} a_{18} = 72 \\ a_{36} = 72 \end{cases} \Rightarrow m = 18$$

Phase 2 :

$$\begin{cases} a_{19} = 34 \\ a_{38} = 46 \end{cases} \begin{cases} a_{20} = 46 \\ a_{40} = 17 \end{cases} \begin{cases} a_{21} = 97 \\ a_{42} = 69 \end{cases} \begin{cases} a_{22} = 17 \\ a_{44} = 24 \end{cases} \begin{cases} a_{23} = 88 \\ a_{46} = 34 \end{cases} \begin{cases} a_{24} = 69 \\ a_{48} = 97 \end{cases}$$

$$\begin{cases} a_{25} = 15 \\ a_{50} = 88 \end{cases}$$

Exemple d'exécution

On pose $E = \{0, 1, 2, \ldots, 100\}$, $f(x) = x^2 + 1 \mod 100$, $a_0 = 3$

Algorithme de Floyd :

Phase 1 :

$$\begin{cases} a_{18} = 72 \\ a_{36} = 72 \end{cases} \Rightarrow m = 18$$

Phase 2 :

$$\begin{cases} a_{19} = 34 \\ a_{38} = 46 \end{cases} \begin{cases} a_{20} = 46 \\ a_{40} = 17 \end{cases} \begin{cases} a_{21} = 97 \\ a_{42} = 69 \end{cases} \begin{cases} a_{22} = 17 \\ a_{44} = 24 \end{cases} \begin{cases} a_{23} = 88 \\ a_{46} = 34 \end{cases} \begin{cases} a_{24} = 69 \\ a_{48} = 97 \end{cases}$$

$$\begin{cases} a_{25} = 15 \\ a_{50} = 88 \end{cases} \begin{cases} a_{26} = 24 \\ a_{52} = 15 \end{cases}$$

Exemple d'exécution

On pose $E = \{0, 1, 2, \ldots, 100\}$, $f(x) = x^2 + 1 \mod 100$, $a_0 = 3$

Algorithme de Floyd :

Phase 1 :

$$\begin{cases} a_{18} = 72 \\ a_{36} = 72 \end{cases} \Rightarrow m = 18$$

Phase 2 :

$$\begin{cases} a_{19} = 34 \\ a_{38} = 46 \end{cases} \begin{cases} a_{20} = 46 \\ a_{40} = 17 \end{cases} \begin{cases} a_{21} = 97 \\ a_{42} = 69 \end{cases} \begin{cases} a_{22} = 17 \\ a_{44} = 24 \end{cases} \begin{cases} a_{23} = 88 \\ a_{46} = 34 \end{cases} \begin{cases} a_{24} = 69 \\ a_{48} = 97 \end{cases}$$

$$\begin{cases} a_{25} = 15 \\ a_{50} = 88 \end{cases} \begin{cases} a_{26} = 24 \\ a_{52} = 15 \end{cases} \begin{cases} a_{27} = 72 \\ a_{54} = 72 \end{cases}$$

Exemple d'exécution

UNIVERSITÉ CAEN NORMANDIE

On pose $E = \{0, 1, 2, \ldots, 100\}$, $f(x) = x^2 + 1 \mod 100$, $a_0 = 3$

Algorithme de Floyd :

Phase 1 :

$$\begin{cases} a_{18} = 72 \\ a_{36} = 72 \end{cases} \Rightarrow m = 18$$

Phase 2 :

$$\begin{cases} a_{19} = 34 \\ a_{38} = 46 \end{cases} \begin{cases} a_{20} = 46 \\ a_{40} = 17 \end{cases} \begin{cases} a_{21} = 97 \\ a_{42} = 69 \end{cases} \begin{cases} a_{22} = 17 \\ a_{44} = 24 \end{cases} \begin{cases} a_{23} = 88 \\ a_{46} = 34 \end{cases} \begin{cases} a_{24} = 69 \\ a_{48} = 97 \end{cases}$$

$$\begin{cases} a_{25} = 15 \\ a_{50} = 88 \end{cases} \begin{cases} a_{26} = 24 \\ a_{52} = 15 \end{cases} \begin{cases} a_{27} = 72 \\ a_{54} = 72 \end{cases} \Rightarrow m_1 = 27$$

Exemple d'exécution

On pose $E = \{0, 1, 2, \ldots, 100\}$, $f(x) = x^2 + 1 \mod 100$, $a_0 = 3$

Algorithme de Floyd :

Phase 1 :

$$\begin{cases} a_{18} = 72 \\ a_{36} = 72 \end{cases} \Rightarrow m = 18$$

Phase 2 :

$$\begin{cases} a_{19} = 34 \\ a_{38} = 46 \end{cases} \begin{cases} a_{20} = 46 \\ a_{40} = 17 \end{cases} \begin{cases} a_{21} = 97 \\ a_{42} = 69 \end{cases} \begin{cases} a_{22} = 17 \\ a_{44} = 24 \end{cases} \begin{cases} a_{23} = 88 \\ a_{46} = 34 \end{cases} \begin{cases} a_{24} = 69 \\ a_{48} = 97 \end{cases}$$

$$\begin{cases} a_{25} = 15 \\ a_{50} = 88 \end{cases} \begin{cases} a_{26} = 24 \\ a_{52} = 15 \end{cases} \begin{cases} a_{27} = 72 \\ a_{54} = 72 \end{cases} \Rightarrow m_1 = 27 \quad \Rightarrow \mu = m_1 - m = 9$$

Exemple d'exécution

On pose $E = \{0, 1, 2, \ldots, 100\}$, $f(x) = x^2 + 1 \mod 100$, $a_0 = 3$

Algorithme de Floyd :

Phase 1 :

$$\begin{cases} a_{18} = 72 \\ a_{36} = 72 \end{cases} \Rightarrow m = 18$$

Phase 2 :

$$\begin{cases} a_{19} = 34 \\ a_{38} = 46 \end{cases} \begin{cases} a_{20} = 46 \\ a_{40} = 17 \end{cases} \begin{cases} a_{21} = 97 \\ a_{42} = 69 \end{cases} \begin{cases} a_{22} = 17 \\ a_{44} = 24 \end{cases} \begin{cases} a_{23} = 88 \\ a_{46} = 34 \end{cases} \begin{cases} a_{24} = 69 \\ a_{48} = 97 \end{cases}$$

$$\begin{cases} a_{25} = 15 \\ a_{50} = 88 \end{cases} \begin{cases} a_{26} = 24 \\ a_{52} = 15 \end{cases} \begin{cases} a_{27} = 72 \\ a_{54} = 72 \end{cases} \Rightarrow m_1 = 27 \quad \Rightarrow \mu = m_1 - m = 9$$

L'algorithme de Floyd est utilisé pour résoudre les problèmes de **factorisation** et de **logarithme discret** en cryptographie.

Génération de nombres premiers

UNIVERSITÉ CAEN NORMANDIE

Que sait-on sur les nombres premiers ?

- Il y a une infinité de nombres premiers
- La proportion de nombres premiers est de plus en plus faible

$$\text{Nb de nombres premiers} \leq n : \quad \pi(n) \sim \frac{n}{\ln n}$$

$$\text{Proportion de nombres premiers} \leq n : \quad \frac{\pi(n)}{n} \sim \frac{1}{\ln n}$$

- Entre deux nombres premiers, l'intervalle peut être aussi grand que l'on veut

Problème : comment générer un nombre premier d'au plus n bits ?

Génération de nombres premiers

UNIVERSITÉ CAEN NORMANDIE

Que sait-on sur les nombres premiers ?

- Il y a une infinité de nombres premiers
- La proportion de nombres premiers est de plus en plus faible

$$\text{Nb de nombres premiers} \leq n : \quad \pi(n) \sim \frac{n}{\ln n}$$

$$\text{Proportion de nombres premiers} \leq n : \quad \frac{\pi(n)}{n} \sim \frac{1}{\ln n}$$

- Entre deux nombres premiers, l'intervalle peut être aussi grand que l'on veut

Problème : comment générer un nombre premier d'au plus n bits ?

```
Gen_Prime(n) :
   p = uniforme(1,2^n)
   tant que p n'est pas premier
      p = uniforme(1,2^n)
   retourner p
```

Loi géométrique et génération de nombres premiers

Proportion de nombres premiers $< 2^n$: $\frac{\pi(2^n)}{2^n} \sim \frac{1}{n \ln 2}$

```
Gen_Prime(n) :
   p = uniforme(1,2^n)
   tant que p n'est pas premier
      p = uniforme(1,2^n)
   retourner p
```

Loi géométrique et génération de nombres premiers

Proportion de nombres premiers$< 2^n$: $\dfrac{\pi(2^n)}{2^n} \sim \dfrac{1}{n \ln 2}$

```
Gen_Prime(n) :
   p = uniforme(1,2^n)
   tant que p n'est pas premier
      p = uniforme(1,2^n)
   retourner p
```

C'est un algorithme de type Las Vegas.
La probabilité de trouver est un nombre premier à chaque étape est $p \sim 1/n \ln 2$.

Loi géométrique et génération de nombres premiers

UNIVERSITÉ CAEN NORMANDIE

Proportion de nombres premiers $< 2^n$: $\dfrac{\pi(2^n)}{2^n} \sim \dfrac{1}{n \ln 2}$

```
Gen_Prime(n) :
   p = uniforme(1,2^n)
   tant que p n'est pas premier
      p = uniforme(1,2^n)
   retourner p
```

C'est un algorithme de type Las Vegas.
La probabilité de trouver est un nombre premier à chaque étape est $p \sim 1/n \ln 2$.

Espérance et variance du nombre d'itérations :

$$\text{espérance} = \frac{1}{p} \sim n \ln 2, \qquad \text{variance} = \frac{1-p}{p^2} \sim (n \ln 2)^2$$

L'algorithme effectue en moyenne un nombre d'itérations linéaire en le nombre de bits (très efficace !)

Loi géométrique et test de connexité

UNIVERSITÉ CAEN NORMANDIE

Processus :

- Soit G_n un graphe à n sommets sans arête.
- Il possède au départ n composantes connexes.
- On ajoute les arêtes une à une aléatoirement et uniformément.

Question : combien d'arêtes faut-il en moyenne pour obtenir un graphe connexe ?

Loi géométrique et test de connexité

UNIVERSITÉ CAEN NORMANDIE

Processus :

- Soit G_n un graphe à n sommets sans arête.
- Il possède au départ n composantes connexes.
- On ajoute les arêtes une à une aléatoirement et uniformément.

Question : combien d'arêtes faut-il en moyenne pour obtenir un graphe connexe ?

Notations :

- Soit X_k : v.a. du nombre d'arêtes ajoutées pour passer de k à $k-1$ CC
- $X = \sum_{k=2}^{n} X_k$ nombre total d'arêtes ajoutées
- A chaque arête ajoutée (i, j), nous avons deux possibilités :
  1. si i et j appartiennent à la même composante alors on ne fait rien
  2. si i et j appartiennent à deux composantes différentes, on fusionne ces composantes (le nombre de composantes diminue).
- Soit p_k la probabilité d'être dans le cas 2. (on passe de k à $k-1$ CC).

Loi géométrique et test de connexité

UNIVERSITÉ CAEN NORMANDIE

Notations :

- Soit X_k : v.a. du nombre d'arêtes ajoutées pour passer de k à $k-1$ CC
- $X = \sum_{k=2}^{n} X_k$ nombre total d'arêtes ajoutées
- A chaque arête ajouté (i, j), nous avons deux possibilités :
  1. si i et j appartiennent à la même composante alors on ne fait rien
  2. si i et j appartiennent à deux composantes différentes, on fusionne ces composantes (le nombre de composantes diminue).
- Soit p_k la probabilité d'être dans le cas 2. (on passe de k à $k-1$ CC).

Pendant la phase k, X_k suit une loi géométrique de paramètre p_k (algorithme Las Vegas).

$$\mathrm{E}[X_k] = \frac{1}{p_k}, \qquad \mathrm{V}(X_k) = \frac{1-p_k}{p_k^2}$$

Loi géométrique et test de connexité

UNIVERSITÉ CAEN NORMANDIE

Notations :

- Soit p_k la probabilité d'être dans le cas 2. (on passe de k à $k-1$ CC).

Pendant la phase k, X_k suit une loi géométrique de paramètre p_k (algorithme Las Vegas).

$$\mathrm{E}[X_k] = \frac{1}{p_k}, \qquad \mathrm{V}(X_k) = \frac{1-p_k}{p_k^2}$$

Minoration de p_k

Soit C_i la composante de i, chaque CC C possède au moins un sommet j, donc cela fait au moins $k-1$ sommets j possibles pour être dans le cas 2. D'où

$$p_k \geq \frac{k-1}{n-1}.$$

Minoration de p_k

$$p_k \geq \frac{k-1}{n-1}.$$

Espérance

$$E[X] = \sum_{k=2}^{n} E[X_k] \leq (n-1) \sum_{k=2}^{n} \frac{1}{k-1} = (n-1)H_{n-1},$$

où H_{n-1} est le $n-1$ème nombre harmonique. On montre que

$$H_n = \ln n + \gamma + \frac{1}{2n} + O(\frac{1}{n^2}),$$

où γ est une constante (la constante d'Euler). Donc $H_n \sim \ln n$. Il faut en moyenne ajouter $n \ln n$ arêtes (en fait moins car nous calculons une majoration) pour obtenir un graphe connexe.

Variance

$$Var(X) = \sum_{k=2}^{n} \frac{1-p_k}{p_k^2} \leq \sum_{k=2}^{n} \frac{(n-k)(n-1)}{(k-1)^2}.$$

$$Var(X) \leq \frac{\pi^2}{6} n^2 - n \ln n.$$

Variance

$$Var(X) = \sum_{k=2}^{n} \frac{1-p_k}{p_k^2} \leq \sum_{k=2}^{n} \frac{(n-k)(n-1)}{(k-1)^2}.$$

$$Var(X) \leq \frac{\pi^2}{6} n^2 - n \ln n.$$

En appliquant l'inégalité de Bienaymé-Chebychev

$$\Pr(|X - E[X]| \geq \frac{\pi n}{\sqrt{6}} t) \leq \frac{1}{t^2},$$

avec $t = \ln \ln n$, il vient

$$\Pr(X \geq n \ln n + n \ln \ln n) = O\Big(\frac{1}{(\ln \ln n)^2}\Big).$$

Loi géométrique et collectionneur de coupons

UNIVERSITÉ CAEN NORMANDIE

Un enfant collectionne des coupons en achetant des paquets de céréales. On suppose qu'il existe n coupons à collectionner et que chaque coupon a la même probabilité d'apparaître dans un paquet.

Problème du Collectionneur de coupons

Combien de paquets l'enfant doit-il acheter en moyenne pour avoir au moins un exemplaire de chacun des coupons ?

Loi géométrique et collectionneur de coupons

UNIVERSITÉ CAEN NORMANDIE

Un enfant collectionne des coupons en achetant des paquets de céréales. On suppose qu'il existe n coupons à collectionner et que chaque coupon a la même probabilité d'apparaître dans un paquet.

Problème du Collectionneur de coupons

Combien de paquets l'enfant doit-il acheter en moyenne pour avoir au moins un exemplaire de chacun des coupons ?

Soit X_i le nombre de paquets achetés pour passer de $i-1$ coupons à i coupons et X le nombre total de paquets achetés.

Q1- Exprimez X en fonction des X_i.

Loi géométrique et collectionneur de coupons

UNIVERSITÉ CAEN NORMANDIE

Un enfant collectionne des coupons en achetant des paquets de céréales. On suppose qu'il existe n coupons à collectionner et que chaque coupon a la même probabilité d'apparaître dans un paquet.

Problème du Collectionneur de coupons

Combien de paquets l'enfant doit-il acheter en moyenne pour avoir au moins un exemplaire de chacun des coupons ?

Soit X_i le nombre de paquets achetés pour passer de $i-1$ coupons à i coupons et X le nombre total de paquets achetés.

Q1- Exprimez X en fonction des X_i.

$$X = X_1 + \ldots + X_n$$

Loi géométrique et collectionneur de coupons

Soit X_i le nombre de paquets achetés pour passer de $i-1$ coupons à i coupons et X le nombre total de paquets achetés.

Q2- A un moment donné, l'enfant possède $i-1$ coupons. Quelle est la probabilité p_i qu'il trouve un nouveau coupon dans un paquet ?

Loi géométrique et collectionneur de coupons

Soit X_i le nombre de paquets achetés pour passer de $i-1$ coupons à i coupons et X le nombre total de paquets achetés.

Q2- A un moment donné, l'enfant possède $i-1$ coupons. Quelle est la probabilité p_i qu'il trouve un nouveau coupon dans un paquet ?

L'enfant trouve un nouveau coupon s'il obtient un coupon parmi les $n-i$ coupons qu'il ne possède pas, comme la distribution est uniforme, on obtient

$$p_i = \frac{n-i+1}{n}.$$

Loi géométrique et collectionneur de coupons

Soit X_i le nombre de paquets achetés pour passer de $i-1$ coupons à i coupons et X le nombre total de paquets achetés.

Q2- A un moment donné, l'enfant possède $i-1$ coupons. Quelle est la probabilité p_i qu'il trouve un nouveau coupon dans un paquet ?

L'enfant trouve un nouveau coupon s'il obtient un coupon parmi les $n-i$ coupons qu'il ne possède pas, comme la distribution est uniforme, on obtient

$$p_i = \frac{n-i+1}{n}.$$

Q3- Quelle est la loi de la variable aléatoire X_i ? (Donner espérance et variance)

Loi géométrique et collectionneur de coupons

Soit X_i le nombre de paquets achetés pour passer de $i-1$ coupons à i coupons et X le nombre total de paquets achetés.

Q2- A un moment donné, l'enfant possède $i-1$ coupons. Quelle est la probabilité p_i qu'il trouve un nouveau coupon dans un paquet ?

L'enfant trouve un nouveau coupon s'il obtient un coupon parmi les $n-i$ coupons qu'il ne possède pas, comme la distribution est uniforme, on obtient

$$p_i = \frac{n-i+1}{n}.$$

Q3- Quelle est la loi de la variable aléatoire X_i ? (Donner espérance et variance)

C'est une loi géométrique de paramètre p_i. $\mathbb{E}[X_i] = \frac{1}{p_i}$, $\mathbb{V}(X_i) = \frac{1-p_i}{p_i^2}$.

Loi géométrique et collectionneur de coupons

Soit X_i le nombre de paquets achetés pour passer de $i-1$ coupons à i coupons et X le nombre total de paquets achetés.

Q2- A un moment donné, l'enfant possède $i-1$ coupons. Quelle est la probabilité p_i qu'il trouve un nouveau coupon dans un paquet ?

L'enfant trouve un nouveau coupon s'il obtient un coupon parmi les $n-i$ coupons qu'il ne possède pas, comme la distribution est uniforme, on obtient

$$p_i = \frac{n-i+1}{n}.$$

Q3- Quelle est la loi de la variable aléatoire X_i ? (Donner espérance et variance)

C'est une loi géométrique de paramètre p_i. $\mathbb{E}[X_i] = \frac{1}{p_i}$, $\mathbb{V}(X_i) = \frac{1-p_i}{p_i^2}$.

Q4- Rappelez la valeur de la somme $\sum_{i=1}^{n} \frac{1}{i}$.

Loi géométrique et collectionneur de coupons

Soit X_i le nombre de paquets achetés pour passer de $i-1$ coupons à i coupons et X le nombre total de paquets achetés.

Q2- A un moment donné, l'enfant possède $i-1$ coupons. Quelle est la probabilité p_i qu'il trouve un nouveau coupon dans un paquet ?

L'enfant trouve un nouveau coupon s'il obtient un coupon parmi les $n-i$ coupons qu'il ne possède pas, comme la distribution est uniforme, on obtient

$$p_i = \frac{n-i+1}{n}.$$

Q3- Quelle est la loi de la variable aléatoire X_i ? (Donner espérance et variance)

C'est une loi géométrique de paramètre p_i. $\mathbb{E}[X_i] = \frac{1}{p_i}$, $\mathbb{V}(X_i) = \frac{1-p_i}{p_i^2}$.

Q4- Rappelez la valeur de la somme $\sum_{i=1}^{n} \frac{1}{i}$.

$\sum_{i=1}^{n} \frac{1}{i} = H_n = \ln n + \gamma + \frac{1}{2n} + O(\frac{1}{n^2})$, le nième nombre harmonique.

Loi géométrique et collectionneur de coupons

Soit X_i le nombre de paquets achetés pour passer de $i-1$ coupons à i coupons et X le nombre total de paquets achetés.

Q5- Calculez l'espérance de X et sa variance.

Loi géométrique et collectionneur de coupons

Soit X_i le nombre de paquets achetés pour passer de $i-1$ coupons à i coupons et X le nombre total de paquets achetés.

Q5- Calculez l'espérance de X et sa variance.

$$\begin{aligned}\mathbb{E}[X] &= \sum_{i=1}^{n} \frac{1}{p_i} \\ &= \sum_{i=1}^{n} \frac{n}{n-i+1} = n \sum_{i=1}^{n} \frac{1}{i} \\ &= nH_n \sim n \ln n.\end{aligned}$$

Loi géométrique et collectionneur de coupons

Soit X_i le nombre de paquets achetés pour passer de $i-1$ coupons à i coupons et X le nombre total de paquets achetés.

Q5- Calculez l'espérance de X et sa variance.

$$\begin{aligned}
\mathbb{E}[X] &= \sum_{i=1}^{n} \frac{1}{p_i} \\
&= \sum_{i=1}^{n} \frac{n}{n-i+1} = n \sum_{i=1}^{n} \frac{1}{i} \\
&= nH_n \sim n \ln n.
\end{aligned}$$

$$\begin{aligned}
\mathbb{V}(X) &= \sum_{i=1}^{n} \mathrm{V}(X_i) = \sum_{i=1}^{n} \frac{1-p_i}{p_i^2} \\
&= \sum_{i=1}^{n} \frac{1}{p_i^2} - \sum_{i=1}^{n} \frac{1}{p_i} \\
&= n^2 \sum_{i=1}^{n} \frac{1}{i^2} - nH_n \\
&\sim \frac{n^2\pi^2}{6} - n \ln n \sim \frac{n^2\pi^2}{6}
\end{aligned}$$

Loi géométrique et collectionneur de coupons

UNIVERSITÉ CAEN NORMANDIE

Q6- Supposons $n = 30$. Combien doit-il acheter de paquets pour avoir moins d'une chance sur 100 de ne pas avoir tous les coupons ?

Loi géométrique et collectionneur de coupons

Q6- Supposons $n = 30$. Combien doit-il acheter de paquets pour avoir moins d'une chance sur 100 de ne pas avoir tous les coupons ?

- En moyenne, on doit collectionner $30 \cdot H_{30} \approx 102$ coupons pour obtenir la collection complète des 30 sortes de coupons.

Loi géométrique et collectionneur de coupons

Q6- Supposons $n = 30$. Combien doit-il acheter de paquets pour avoir moins d'une chance sur 100 de ne pas avoir tous les coupons ?

- En moyenne, on doit collectionner $30 \cdot H_{30} \approx 102$ coupons pour obtenir la collection complète des 30 sortes de coupons.
- La variance vaut $\mathbb{V}(X) \sim \frac{n^2\pi^2}{6} - n \ln n \approx 1376$

Loi géométrique et collectionneur de coupons

Q6- Supposons $n = 30$. Combien doit-il acheter de paquets pour avoir moins d'une chance sur 100 de ne pas avoir tous les coupons ?

- En moyenne, on doit collectionner $30 \cdot H_{30} \approx 102$ coupons pour obtenir la collection complète des 30 sortes de coupons.
- La variance vaut $\mathbb{V}(X) \sim \frac{n^2\pi^2}{6} - n \ln n \approx 1376$
- Selon l'inégalité de Bienaymé-Chebychev

$$\mathbb{P}\left(|X - \mathbb{E}[X]| \geq t\right) \leq \frac{\mathbb{V}(X)}{t^2}$$

Loi géométrique et collectionneur de coupons

Q6- Supposons $n = 30$. Combien doit-il acheter de paquets pour avoir moins d'une chance sur 100 de ne pas avoir tous les coupons ?

- En moyenne, on doit collectionner $30 \cdot H_{30} \approx 102$ coupons pour obtenir la collection complète des 30 sortes de coupons.
- La variance vaut $\mathbb{V}(X) \sim \frac{n^2\pi^2}{6} - n \ln n \approx 1376$
- Selon l'inégalité de Bienaymé-Chebychev

$$\mathbb{P}\left(|X - \mathbb{E}[X]| \geq t\right) \leq \frac{\mathbb{V}(X)}{t^2}$$

- En prenant $t = \sqrt{100\mathbb{V}(X)}$ on obtient

$$\mathbb{P}\left(|X - \mathbb{E}[X]| \geq t\right) \leq \frac{1}{100}$$

Loi géométrique et collectionneur de coupons

UNIVERSITÉ CAEN NORMANDIE

Q6- Supposons $n = 30$. Combien doit-il acheter de paquets pour avoir moins d'une chance sur 100 de ne pas avoir tous les coupons ?

- En moyenne, on doit collectionner $30 \cdot H_{30} \approx 102$ coupons pour obtenir la collection complète des 30 sortes de coupons.
- La variance vaut $\mathbb{V}(X) \sim \frac{n^2\pi^2}{6} - n \ln n \approx 1376$
- Selon l'inégalité de Bienaymé-Chebychev

$$\mathbb{P}\left(|X - \mathbb{E}[X]| \geq t\right) \leq \frac{\mathbb{V}(X)}{t^2}$$

- En prenant $t = \sqrt{100\mathbb{V}(X)}$ on obtient

$$\mathbb{P}\left(|X - \mathbb{E}[X]| \geq t\right) \leq \frac{1}{100}$$

- soit

$$\mathbb{P}\left(X \geq \mathbb{E}[X] + t\right) = \mathbb{P}\left(X \geq 102 + 371\right) \leq \frac{1}{100}$$

Loi géométrique et collectionneur de coupons

UNIVERSITÉ CAEN NORMANDIE

Q7- Comment utilisez ce problème pour tester un générateur pseudo-aléatoire ?

Loi géométrique et collectionneur de coupons

UNIVERSITÉ CAEN NORMANDIE

Q7- Comment utilisez ce problème pour tester un générateur pseudo-aléatoire ?

1. On découpe la suite aléatoire en blocs de taille m (par exemple, $m = 10$.
2. On a alors une collection de $n = 2^{10} = 1024$ coupons (chaque mot binaire de longueur m est un coupon).
3. On regarde si le nombre de blocs nécessaire pour avoir la collection complète se rapproche de $\mathbb{E}[X]$.
4. On peut aussi le faire un grand nombre de fois et regarder la répartition.

QuickSort et randomQuickSort

UNIVERSITÉ CAEN NORMANDIE

```
partition(tableau T, entier début, entier fin, entier pivot)
échanger T[pivot] et T[fin]
  j  := début
  pour i de début à fin - 1
  si T[i] <= T[fin] alors
   échanger T[i] et T[j]
   j  := j + 1
  échanger T[fin] et T[j]
  renvoyer j


Quicksort(tableau T, entier début, entier fin)
    si début < fin alors
        pivot  := début
        pivot  := partition(T, début, fin, pivot)
        Quicksort(T, début, pivot-1)
        Quicksort(T, pivot+1, fin)
    fin si

RandomQuicksort(tableau T, entier début, entier fin)
    si début < fin alors
        pivot  := RandomPivot(T,début,fin)
        pivot  := partition(T, début, fin, pivot)
        RandomQuicksort(T, début, pivot-1)
        RandomQuicksort(T, pivot+1, fin)
    fin si
```

Remarques sur Quicksort

UNIVERSITÉ CAEN NORMANDIE

- C'est un algorithme déterministe
- Pire des cas : quand le tableau est trié !
- Complexité dans le pire de cas : $O(n^2)$ comparaisons
- Dans le modèle des permutations aléatoires, la complexité moyenne de Quicksort vérifie

$$C_n = n - 1 + \frac{1}{n}\sum_{i=0}^{n-1}(C_i + C_{n-i-1}) \sim 2n \ln n.$$

En moyenne, QuickSort est quasi-linéaire !

Remarques sur Quicksort

UNIVERSITÉ CAEN NORMANDIE

- C'est un algorithme déterministe
- Pire des cas : quand le tableau est trié !
- Complexité dans le pire de cas : $O(n^2)$ comparaisons
- Dans le modèle des permutations aléatoires, la complexité moyenne de Quicksort vérifie

$$C_n = n - 1 + \frac{1}{n}\sum_{i=0}^{n-1}(C_i + C_{n-i-1}) \sim 2n \ln n.$$

En moyenne, QuickSort est quasi-linéaire !

Mais cette différence peut être un problème dans certaines applications, surtout celles proches du pire des cas ...

Etude de RandomQuickSort

UNIVERSITÉ CAEN NORMANDIE

Soit C la v.a. du nombre de comparaisons de RandomQuickSort.

Soient i et $j \in \{1, \ldots, n\}$ avec $i < j$. Soit $C_{i,j}$ la v.a définie par

$$\begin{aligned} C_{i,j} &= 1 \text{ lorsque } i \text{ et } j \text{ sont comparés entre-eux} \\ &= 0 \text{ sinon} \end{aligned}$$

Etude de RandomQuickSort

Soit C la v.a. du nombre de comparaisons de RandomQuickSort.

Soient i et $j \in \{1, \ldots, n\}$ avec $i < j$. Soit $C_{i,j}$ la v.a définie par

$$\begin{aligned} C_{i,j} &= 1 \text{ lorsque } i \text{ et } j \text{ sont comparés entre-eux} \\ &= 0 \text{ sinon} \end{aligned}$$

Il vient

$$C = \sum_{i,j \in \{1,\ldots,n\}, i<j} C_{i,j}.$$

Etude de RandomQuickSort

Soit C la v.a. du nombre de comparaisons de RandomQuickSort.

Soient i et $j \in \{1, \ldots, n\}$ avec $i < j$. Soit $C_{i,j}$ la v.a définie par

$$\begin{array}{rcl} C_{i,j} & = & 1 \text{ lorsque } i \text{ et } j \text{ sont comparés entre-eux} \\ & = & 0 \text{ sinon} \end{array}$$

Il vient

$$C = \sum_{i,j \in \{1,\ldots,n\}, i<j} C_{i,j}.$$

Notons

$$p_{i,j} = \mathbb{P}(C_{i,j} = 1) = \mathbb{E}[C_{i,j}].$$

Il vient

$$\mathbb{E}[C] = \sum_{i,j \in \{1,\ldots,n\}, i<j} \mathbb{E}[C_{i,j}] = \sum_{i,j \in \{1,\ldots,n\}, i<j} p_{i,j}.$$

Il reste donc à calculer $p_{i,j}$! ! !

Etude de RandomQuickSort

UNIVERSITÉ CAEN NORMANDIE

Il reste donc à calculer $p_{i,j}$! ! !

- Au début, i et j sont dans les mêmes parties du tableau et peuvent être comparés
- Si i ou j est choisi comme pivot, alors i et j sont comparés 1 fois, et ne seront plus comparés ensuite
- Si le pivot est choisi dans l'intervalle $]i, j[$, alors i et j ne seront plus dans la même partie du tableau et ne seront plus comparés

La probabilité du deuxième cas est donnée par

Etude de RandomQuickSort

UNIVERSITÉ CAEN NORMANDIE

Il reste donc à calculer $p_{i,j}$! ! !

- Au début, i et j sont dans les mêmes parties du tableau et peuvent être comparés
- Si i ou j est choisi comme pivot, alors i et j sont comparés 1 fois, et ne seront plus comparés ensuite
- Si le pivot est choisi dans l'intervalle $]i, j[$, alors i et j ne seront plus dans la même partie du tableau et ne seront plus comparés

La probabilité du deuxième cas est donnée par $p_{i,j} = \dfrac{2}{j - i + 1}$

Etude de RandomQuickSort

UNIVERSITÉ CAEN NORMANDIE

Il reste donc à calculer $p_{i,j}$! ! !

- Au début, i et j sont dans les mêmes parties du tableau et peuvent être comparés
- Si i ou j est choisi comme pivot, alors i et j sont comparés 1 fois, et ne seront plus comparés ensuite
- Si le pivot est choisi dans l'intervalle $]i, j[$, alors i et j ne seront plus dans la même partie du tableau et ne seront plus comparés

La probabilité du deuxième cas est donnée par $p_{i,j} = \dfrac{2}{j-i+1}$

$$\begin{aligned} \mathbb{E}[C] &= \sum_{(i,j)\in\{1\leq i<j\leq n\}} \frac{2}{j-i+1} \\ &= \dots \\ &= 2nH_n - 2n + 3 \sim 2n \ln n \end{aligned}$$

$\rightarrow$ C'est la même asymptotique que pour la complexité en moyenne de QuickSort !

Etude de RandomQuickSort

UNIVERSITÉ CAEN NORMANDIE

Il reste donc à calculer $p_{i,j}$! ! !

- Au début, i et j sont dans les mêmes parties du tableau et peuvent être comparés
- Si i ou j est choisi comme pivot, alors i et j sont comparés 1 fois, et ne seront plus comparés ensuite
- Si le pivot est choisi dans l'intervalle $]i, j[$, alors i et j ne seront plus dans la même partie du tableau et ne seront plus comparés

La probabilité du deuxième cas est donnée par $p_{i,j} = \frac{2}{j-i+1}$

$$\begin{aligned} \mathbb{E}[C] &= \sum_{(i,j)\in\{1\leq i<j\leq n\}} \frac{2}{j-i+1} \\ &= \dots \\ &= 2nH_n - 2n + 3 \sim 2n \ln n \end{aligned}$$

$\rightarrow$ C'est la même asymptotique que pour la complexité en moyenne de QuickSort !
$\rightarrow$ Mais la méthode garantit une complexité en moyenne quasi-linéaire quelle que soit l'instance. Il n'y a donc "plus" de pire des cas ou de meilleur des cas.

Classes de complexité probabilistes

UNIVERSITÉ CAEN NORMANDIE

Une classe de complexité regroupe les problèmes (langages) qui partagent une même propriété en lien avec le temps d'exécution ou l'espace mémoire requis pour les résoudre.

Nous parlerons ici de problèmes de décision mais en Théorie de la complexité, on préfère souvent parler de langage reconnaissable par un modèle de calcul (machines de Turing, machines RAM, automates cellulaires, machines quantiques etc.).

Lien entre les deux points de vue :

Problème de décision	Langage
Instances du problème	Langage $\mathcal{I} \subset \{0,1\}^\star$
Instance positives	Langage $\mathcal{I}^+ \subset \{0,1\}^\star$
Résoudre le problème	Reconnaître le langage $\mathcal{I}^+$

Reconnaître si $w \in \{0,1\}^\star$ (ou $\mathcal{I}$) est dans le langage $\mathcal{I}^+$ revient à résoudre le problème de décision (beaucoup de détails sont omis dans cette description).

Classes de complexité probabilistes

UNIVERSITÉ CAEN NORMANDIE

Definition

Un problème de décision est dit dans la classe RP (random polynomial-time) s'il existe un algorithme probabiliste $\mathcal{A}$ tel que

$$\begin{aligned} x \text{ est une instance positive} &\Rightarrow Pr\left[\mathcal{A}(x) = 1\right] \geq \frac{1}{2} \\ x \text{ est une instance négative} &\Rightarrow Pr\left[\mathcal{A}(x) = 1\right] = 0 \end{aligned}$$

Question 1. Pour les problèmes dans RP, quelle réponse (0 ou 1) ne donne lieu à aucune erreur ? Justifiez.

Question 2. Donnez un exemple de problème vu en cours qui est dans RP.

Classes de complexité probabilistes

UNIVERSITÉ CAEN NORMANDIE

Definition

Un problème de décision est dit dans la classe $co - RP$ (complementary random polynomial-time) s'il existe un algorithme probabiliste $\mathcal{A}$ tel que

$$x \text{ est une instance positive} \quad \Rightarrow \quad Pr\left[\mathcal{A}(x) = 1\right] = 1$$

$$x \text{ est une instance négative} \quad \Rightarrow \quad Pr\left[\mathcal{A}(x) = 0\right] \geq \frac{1}{2}$$

Question 3. Pour les problèmes dans $co - RP$, quelle réponse (0 ou 1) ne donne lieu à aucune erreur ? Justifiez.

Question 4. Donnez un exemple de problème vu en cours qui est dans $co - RP$.

Classes de complexité probabilistes

UNIVERSITÉ CAEN NORMANDIE

Definition

Un problème de décision est dit dans la classe $co - RP$ (complementary random polynomial-time) s'il existe un algorithme probabiliste $\mathcal{A}$ tel que

$$x \text{ est une instance positive} \quad \Rightarrow \quad Pr\left[\mathcal{A}(x) = 1\right] = 1$$

$$x \text{ est une instance négative} \quad \Rightarrow \quad Pr\left[\mathcal{A}(x) = 0\right] \geq \frac{1}{2}$$

Question 5. Que pensez-vous de la valeur $1/2$ dans les définitions précédentes ? Peut-elle être remplacée par une autre valeur ? Si oui, comment procéder ?

Classes de complexité probabilistes

UNIVERSITÉ CAEN NORMANDIE

Definition

Pour un problème de décision $\mathcal{D}$ et une instance x du problème, on note $\chi_{\mathcal{D}}(x)$ la réponse (1=positive ou 0=négative) au problème. Un problème de décision $\mathcal{D}$ est dit dans la classe *ZPP* (Zero error probability) s'il existe un algorithme probabiliste $\mathcal{A}$, de complexité (dans le pire des cas) polynomiale, retournant 0, 1 ou *FAIL* tel que

$$\forall x, \qquad Pr\left[\mathcal{A}(x) = FAIL\right] \leq \frac{1}{2}$$

$$\forall x \quad Pr\left[\mathcal{A}(x) = \chi_{\mathcal{D}}(x) \text{ ou } \mathcal{A}(x) = FAIL\right] = 1$$

Question 6. Considérons un problème dans *ZPP* et un algorithme probabiliste $\mathcal{A}$ pour le résoudre. Construire un algorithme $\mathcal{B}$ qui n'échoue jamais et qui retourne toujours la bonne réponse.

Question 7. Soit $p(n)$ la complexité de l'algorithme $\mathcal{A}$ sur des instances de taille n. Donnez une borne pour la complexité moyenne de l'algorithme $\mathcal{B}$.

Question 8. A quelle famille d'algorithmes appartient l'algorithme $\mathcal{B}$?

Classes de complexité probabilistes

UNIVERSITÉ CAEN NORMANDIE

Definition

ZPP = soit FAIL avec proba $\leq 1/2$, soit la bonne réponse

Question 9. Donnez un exemple de problème dans ZPP qui a été vu en cours.

Question 10. Montrez que $ZPP \subseteq RP$.

Question 11. Montrez que $ZPP \subseteq co - RP$.

Question 12. Montrez que $ZPP = RP \cap co - RP$.

Autres algorithmes probabilistes

UNIVERSITÉ CAEN NORMANDIE

Nous verrons en TD/TP ou plus tard dans le cours d'autres algorithmes probabilistes

- Sélection de la médiane
- Problème des n reines
- Générateurs pseudo-aléatoires
- Paradoxe des anniversaires et fonctions de hachage
- protocoles cryptographiques probabilistes
- ...

Autres algorithmes probabilistes

UNIVERSITÉ CAEN NORMANDIE

Nous verrons en TD/TP ou plus tard dans le cours d'autres algorithmes probabilistes

- Sélection de la médiane
- Problème des n reines
- Générateurs pseudo-aléatoires
- Paradoxe des anniversaires et fonctions de hachage
- protocoles cryptographiques probabilistes
- ...

Nous n'aborderons pas certaines familles d'algorithmes probabilistes

- à base de chaînes de Markov (Récuit Simulé)
- à base de statistiques

Autres algorithmes probabilistes

UNIVERSITÉ CAEN NORMANDIE

Nous verrons en TD/TP ou plus tard dans le cours d'autres algorithmes probabilistes

- Sélection de la médiane
- Problème des n reines
- Générateurs pseudo-aléatoires
- Paradoxe des anniversaires et fonctions de hachage
- protocoles cryptographiques probabilistes
- ...

Nous n'aborderons pas certaines familles d'algorithmes probabilistes

- à base de chaînes de Markov (Récuit Simulé)
- à base de statistiques

Nous verrons aussi en TD un peu de théorie de la complexité