IFT2125 - Introduction à l'algorithmique

Algorithmes probabilistes (B&B chapitre 10)

Pierre McKenzie

DIRO, Université de Montréal

Automne 2017

Characteristics of probabilistic algorithms

Caractéristiques des algorithmes probabilistes

B&B section 10.1

play toss

- jouent à pile ou face
- se comportent différemment, exécutés deux fois sur le même exemplaire Behave differently, run twice on the same copy
- peuvent se tromper can be wrong
- défient parfois l'intuition.

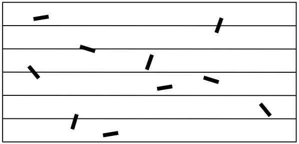
sometimes defy intuition.

Première surprise First surprise

Le hasard peut être utile
Chance can be useful

create a solar system (here still waiting for a verdict)

- créer un système solaire (ici toujours en attente de verdict)
- estimer π
  estimate



allow some cryptographic protocols

- permettre certains protocoles cryptographiques
- vérifier la primalité rapidement reduce the effect of bad copies.
- accélérer une recherche speed up a search
- réduire l'effet de mauvais exemplaires.
  check the primality quickly

Deuxième surprise Second surprise

Le hasard peut être précis
Chance can be precise

Exemple : pile=succès, face=échec
Example: stack = success, face = failure

a success, after a test
- Pr[un succès, après un essai] $= \frac{1}{2}$
- Pr[un succès ou plus, après 2 essais] $=$
a success or more after 2 attempts

Le hasard peut être précis
Chance can be precise

Exemple : pile=succès, face=échec
Example: stack = success, face = failure

a success, after a test
- Pr[un succès, après un essai] $= \frac{1}{2}$
- Pr[un succès ou plus, après 2 essais] $= \frac{3}{4}$
a success or more after 2 attempts
- Pr[un succès ou plus, après 3 essais] $=$
a success or more after 3 attempts

Deuxième surprise Second surprise

Le hasard peut être précis
Chance can be precise

Exemple : pile=succès, face=échec
Example: stack = success, face = failure

- a success, after a test
  Pr[un succès, après un essai] $= \frac{1}{2}$
- Pr[un succès ou plus, après 2 essais] $= \frac{3}{4}$
  a success or more after 2 attempts
- Pr[un succès ou plus, après 3 essais] $= \frac{7}{8}$
  a success or more after 3 attempts
- ...
- Pr[un succès ou plus, après n essais] $=$
  a success or more after n attempts

Deuxième surprise Second surprise

Le hasard peut être précis
Chance can be precise

Exemple : pile=succès, face=échec
Example: stack = success, face = failure

a success, after a test
- Pr[un succès, après un essai] $= \frac{1}{2}$
- Pr[un succès ou plus, après 2 essais] $= \frac{3}{4}$
a success or more after 2 attempts
- Pr[un succès ou plus, après 3 essais] $= \frac{7}{8}$
- ...
- Pr[un succès ou plus, après n essais] $= 1 - (\frac{1}{2})^n$
a success or more after n attempts

$\Rightarrow$ une suite de 1000 échecs consécutifs est moins probable qu'une erreur interne de l'ordinateur après une seconde de calcul !
a sequence of 1000 consecutive failures is less likely than an internal computer error after one second of calculation !

Trois types d'algorithmes probabilistes

B&B section 10.2 Three types of probabilistic algorithms

Numerical
- Numérique

  approximate solution to a numerical problem (eg simulation)
  - solution approximative à un problème numérique (ex : simulation)
  - plus de temps ⇒ plus de précision.

    more time ⇒ more precision

- Monte Carlo

  always an answer (ex: yes or no)
  - toujours une réponse (ex : oui ou non)

  often impossible to effectively check the answer
  - souvent impossible de vérifier efficacement la réponse
  - plus de temps ⇒ meilleure proba de bonne réponse.

    more time ⇒ better proba of correct answer.

- Las Vegas

  never an inaccurate answer, but sometimes without answer
  - jamais de réponse inexacte, mais parfois sans réponse
  - plus de temps ⇒ meilleure proba de réponse.

    more time ⇒ better response probability

Three types

Trois types

Quand Christophe Colomb a-t-il atteint l'Amérique ?

When did Christopher Columbus reach America?

Numerical

- Numérique

  Au 15ième siècle In the 15th century

  entre 1493 et 1499 between 1493 and 1499

  entre 1489 et 1496 between 1489 and 1496

- Monte Carlo

  1492, 1501, 567, 765, 1492, 1487, 1488, 1501, 1500, ...

- Las Vegas

  1492, 1492, sais pas, sais pas, 1492, sais pas, 1492, 1492, ...

  do not know

Average time vs. expected time

Temps moyen vs. temps espéré

B&B section 10.3

Recall

- Rappel :

time

$$t_{\text{moyen}}(n) = \frac{\sum_{|w|=n} \text{temps}(w)}{\#\{w : |w| = n\}}$$

average

Expected time first defined on each copy

- Temps espéré d'abord défini sur chaque exemplaire :

time with

$$t_{\text{espéré}}(w) = \sum_{\text{suites } \sigma \text{ de piles/faces menant à l'arrêt}} (\text{temps}(w \text{ avec } \sigma)) \times \Pr[\sigma]$$

expected

σ stacks / faces leading to shutdown

then

- puis $t_{\text{espéré}}(n) = \max_{|w|=n} t_{\text{espéré}}(w)$

expected expected

How to generate m bits (pseudo-) random?

Comment générer m bits (pseudo-) aléatoires ?

- pas si simple
  Not that easy
- a possible method
  une méthode possible :
  - choose p and q first two ≡ 3 mod 4 a hundred digits
    choisir p et q deux premiers $\equiv 3 \mod 4$ d'une centaine de chiffres
  - former entier z de 200 chiffres en utilisant l'heure en pico-secondes
    develop integer z of 200 digits using the time in pico-seconds
  - vérifier que $\text{pgcd}(z, pq) = 1$
    check that pgcd (z, pq) = 1
  - pour $i \leftarrow 1$ à m faire [$z \leftarrow z \times z \mod pq$; imprimer parité(z)]
    for i ← 1 to make [z ← z × z mod pq; print parity (z)]
- suite obtenue presque toujours indistingable d'une suite aléatoire, même si la suite se répétera à coup sûr si m très grand
- cf. Pierre L'Écuyer du DIRO

result obtained almost always indistinguishable from a random sequence, even if the following will be repeated for sure if m very large

Intégration numérique, B&B section 10.5.2
numerical integration

- Pour estimer $I = \int_a^b f(x)\mathrm{d}x$, l'idée : the idea
  To estimate
  - estimer la hauteur $I/(b-a)$ du rectangle
    estimate the height I / (b - a) of the rectangle

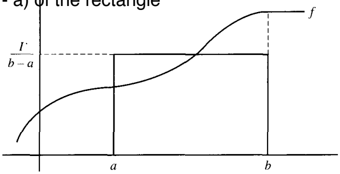


Figure 10.2. Numerical integration

  multiply by b - a.
  - multiplier par $b-a$.
- Solution déterministe : Deterministic solution:
  take m equidistant points between a and b inclusively
  - prendre m points équidistants entre a et b inclusivement
  - évaluer f à chacun de ces points evaluate f at each of these points
  - prendre la moyenne, voilà $I/(b-a)$.
    take the average, that's I / (b - a).
- Solution probabiliste :
  - engendrer les m points entre a et b au hasard
    generate the m points between a and b at random
- Quelle méthode est la meilleure ?
  Which method is the best?

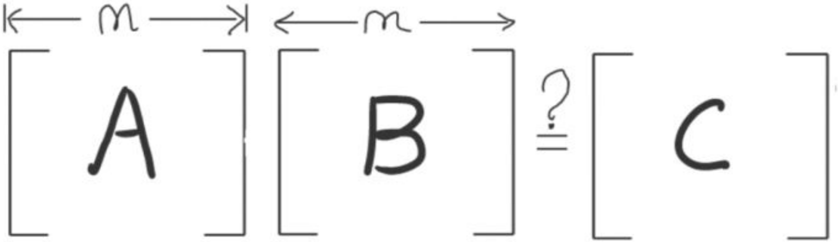
always answer
- répond toujours
- peut se tromper can be wrong
- aucun avertissement en cas d'erreur no warning in case of error
- mais réussit avec bonne probabilité sur tout exemplaire
  but succeeds with good probability on any copy

L'algo est *p-correct*, $0 < p < 1$, si $\Pr[\text{bonne réponse}] \geq p$.
The algo is p-correct, 0 < p < 1, if Pr [correct answer] ≥ p.

Monte Carlo : exemple 1

Vérifier en $O(n^2)$ que $AB = C$, B&B section 10.6.1

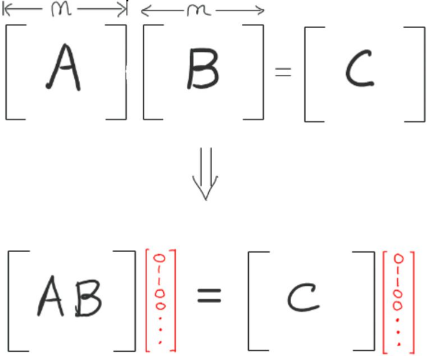
Check in O (n^2) that AB = C, B & B section 10.6.1



L'idée the idea

Choisir $X \in \{0,1\}^n$ au hasard et exploiter

Choose X ∈ {0, 1} n at random and exploit



by answering yes if this last identity is verified, no otherwise.

en répondant oui si cette dernière identité est vérifée, non sinon.

- Alors pas d'erreur possible lorsque $AB = C$: cool !
- Mais quelle probabilité d'erreur si $AB \neq C$?
- Et comment vérifier cette dernière identité en $O(n^2)$?

So no mistake possible when AB = C: cool!
But what probability of error if AB != C?
And how to check this last identity in O (n^2)?

Mais quelle probabilité d'erreur si $AB \neq C$?

But what probability of error if AB! = C?

$$\begin{bmatrix} AB - C \end{bmatrix} \overbrace{\begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ \vdots \end{bmatrix}}^{\text{au hasard}} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

$\exists \bullet_{ij} = d_{ij} \neq 0$ et écrivons r_j et $p_i = d_{ij} r_j + y$

Suite en classe. On obtient : $\leq \frac{1}{2}$.

Suite in class. We obtain :

Monte Carlo : exemple 1

(suite)
cont.

In summary :

En résumé :

on copy / error
- sur exemplaire $AB = C$, $\Pr[\text{erreur}] = 0$

on copy. / error
- sur exemplaire $AB \neq C$, $\Pr[\text{erreur}] \leq \frac{1}{2}$

in all cases / error
- dans tous les cas, $\Pr[\text{erreur}] \leq \frac{1}{2}$

as a bonus, biased because no error on positive copies
- en prime, biaisé car aucune erreur sur exemplaires positifs
  - plus précisément, faux-biaisé car si l'algo répond "faux" il ne se trompe jamais, i.e., $AB \neq C$.

    more precisely, false-biased because if the algo answers "false" it is not wrong
    never, i.e., AB != C.

Monte Carlo : exemple 1

(suite)
cont.

Can we reduce the error by repeating k times? Let's see:

Peut-on réduire l'erreur en répétant k fois ? Voyons :

- faux-biaisé $\Rightarrow$
  false biased

Monte Carlo : exemple 1

(suite)
cont.

Can we reduce the error by repeating k times? Let's see:

Peut-on réduire l'erreur en répétant k fois ? Voyons :

- faux-biaisé $\Rightarrow$
  false biased
  we can conclude from the first answer "false"
  - on peut conclure dès la première réponse "faux"
  - on ne répondra "vrai" qu'après k réponses "vrai"
    we will answer "true" only after "true" answers
- l'erreur sur exemplaire $AB = C$?
  the error on copy

Monte Carlo : exemple 1

(suite)
cont.

Can we reduce the error by repeating k times? Let's see:

Peut-on réduire l'erreur en répétant k fois ? Voyons :

- faux-biaisé $\Rightarrow$
  false biased
  we can conclude from the first answer "false"
  - on peut conclure dès la première réponse "faux"
  - on ne répondra "vrai" qu'après k réponses "vrai"
    we will answer "true" only after "true" answers
- l'erreur sur exemplaire $AB = C$?
  the error on copy
  - toutes les réponses seront vrai   all the answers will be true
  - $\Pr[\text{erreur}] = 0$
- l'erreur sur exemplaire $AB \neq C$?
  the error on copy

Monte Carlo : exemple 1

(suite)
cont.

Peut-on réduire l'erreur en répétant k fois ? Voyons :

- faux-biaisé $\Rightarrow$
  false biased
  we can conclude from the first answer "false"
  - on peut conclure dès la première réponse "faux"
  - on ne répondra "vrai" qu'après k réponses "vrai"
  we will answer "true" only after "true" answers
- l'erreur sur exemplaire $AB = C$?
  the error on copy
  - toutes les réponses seront vrai all the answers will be true
  - $\Pr[\text{erreur}] = 0$
- l'erreur sur exemplaire $AB \neq C$?
  the error on copy
  - scénario semblable à "pile=succès" et "face=échec"
  scenario similar to "stack = success" and "face = failure"
  - $\Pr[k \text{ échecs consécutifs}] \leq (\frac{1}{2})^k$
  k consecutive failures
- dans tous les cas, $\Pr[\text{erreur}] \leq (\frac{1}{2})^k$
  in all cases
- l'algorithme répété k fois de cette façon est $1 - (\frac{1}{2})^k$-correct
  the algorithm repeated k times this way is 1-(1/2)^k correct

Monte Carlo : exemple 2

Test d'identité de polynômes, n'est pas dans B&B

Identity test of polynomials, is not in B & B

TIP

given two polynomials described by arithmetic circuit

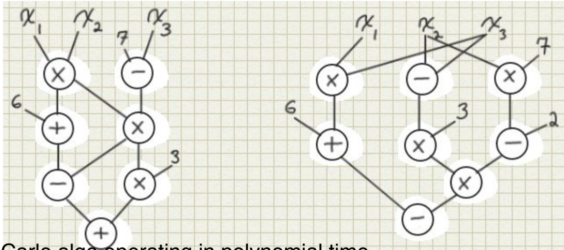
DONNÉE: deux polynômes décrits par circuit arithmétique

DÉCIDER: si ces deux polynômes sont identiques

decide if these two polynomials are identical

Exemple :

Example



In demo: a Monte Carlo algo operating in polynomial time.

Notorious fact: no non-probabilistic polynomial algorithm solving this problem is known at present

En démo : un algo de Monte Carlo fonctionnant en temps polynomial.

Fait notoire : aucun algorithme polynomial non probabiliste résolvant ce problème n'est connu à l'heure actuelle.

Monte Carlo : exemple 3

Test de primalité, B&B section 8.6.2
Primality test

<u>PRIMALITÉ</u>

given an integer m in binary
DONNÉE: un entier m en binaire

DÉCIDER: si m est premier
decide if m is prime

a Monte Carlo algo in O (n3) exists (n = log2 m)
- un algo de Monte Carlo en $O(n^3)$ existe $(n = \log_2 m)$
- m est premier $\rightarrow \Pr[\text{erreur}] = 0$
  m is prime
- m n'est pas premier $\rightarrow \Pr[\text{erreur}] \leq \frac{1}{4}$
  m is not prime
- amplifier la probabilité de succès par répétitions est donc possible
  amplify the probability of success by repetitions is possible

Cet algo est faux-biaisé ou vrai-biaisé ?
This algo is false-biased or true-biased?

Monte Carlo : exemple 3

Test de primalité, B&B section 8.6.2
Primality test

<u>PRIMALITÉ</u>
given an integer m in binary
DONNÉE: un entier m en binaire

DÉCIDER: si m est premier
decide if m is prime

a Monte Carlo algo in O (n3) exists (n = log2 m)
- un algo de Monte Carlo en $O(n^3)$ existe $(n = \log_2 m)$
- m est premier $\rightarrow \Pr[\text{erreur}] = 0$
  m is prime
- m n'est pas premier $\rightarrow \Pr[\text{erreur}] \leq \frac{1}{4}$
  m is not prime
- amplifier la probabilité de succès par répétitions est donc possible
  amplify the probability of success by repetitions is possible

This algo is false-biased or true-biased?
Cet algo est faux-biaisé ou vrai-biaisé ?

Faux-biaisé car ne répond jamais "non" quand le nombre est premier
False-biased because never answers "no" when the number is prime

Monte Carlo : exemple 3

Primalité (suite)

primality (cont.)

a non-probabilistic polynomial algo exists only since 2002

- un algo polynomial non probabiliste n'existe que depuis 2002
- l'algo est compliqué the algo is complicated
- il requiert temps $\Omega(n^5)$ it requires time Ω (n5)
- l'algo de Monte Carlo est toujours utilisé dans la pratique

  the algo of Monte Carlo is still used in practice

Monte Carlo

Quand peut-on amplifier l'avantage stochastique ? (B&B section 10.6.4)
When can one amplify the stochastic advantage?

Two cases
Deux cas :

- Algo biaisé Algo biased
- Algo non biaisé Algo unbiased

Biased case

Cas biaisé

Exemple : un algo $A(x)$ à réponse vrai/faux, vrai-biaisé et $\frac{3}{4}$-correct

Example: an algo A (x) with true / false response, true-biased and 3/4-correct

How to boost this 3/4?

Comment amplifier ce $\frac{3}{4}$?

Biased case
Cas biaisé

Exemple : un algo $A(x)$ à réponse vrai/faux, vrai-biaisé et $\frac{3}{4}$-correct
Example: an algo A (x) with true / false response, true-biased and 3/4-correct

How to boost this 3/4?
Comment amplifier ce $\frac{3}{4}$?

Déjà vu :
Already seen

```
fonction ampli_biaisée(x,k)
    pour i = 1 à k faire   do
    for
        if si A(x) alors   then
            retourner vrai
                           return true
    retourner faux
           return false
```

Autrement dit : on s'arrête dès le premier **vrai**, sans quoi on répond **faux**.
In other words: we stop at the first true, otherwise we answer false.

Cet exemple, avec $k = 3$ répétitions

This example, with k = 3 repetitions

On a copy x "false"

- Sur un exemplaire x "faux" :
  A (x) will never answer "true" because A (x) true-biased
  - $A(x)$ ne répondra jamais "vrai" car $A(x)$ vrai-biaisé
  - ampli_biaisée(x,3) concluera "faux" will conclude "false"
  - Pr[ampli_biaisée(x,3) se trompe] $= 0$.
    is mistaken

On a copy x "true"

- Sur un exemplaire x "vrai"
  only possibility of error = A (x) answers "false" 3 times
  - seule possibilité d'erreur $= A(x)$ répond "faux" 3 fois
  - Pr[err, err, err] $= \frac{1}{4} \times \frac{1}{4} \times \frac{1}{4}$
  - Pr[ampli_biaisée(x,3) se trompe] $= \frac{1}{64} \approx 2\%$.
    is mistaken

- Passé de 75%-correct à 98%-correct

Même exemple $A(x)$, mais cette fois $p = \frac{1}{100}$-correct

Same example A (x), but this time

Here A (x) is wrong 99 times out of 100, yet ...

- Ici $A(x)$ se trompe 99 fois sur 100, pourtant...

... amplifying is possible because the only possibility of error of A (x) is always to answer "false" on a true copy:

- ...amplifier est possible, car seule possibilité d'erreur de $A(x)$ est toujours de répondre "faux" sur un exemplaire vrai :
  - $\Pr[k \text{ erreurs consécutives de } A(x)] = \left(\frac{99}{100}\right)^k$

    K consecutive errors of A(x)

  - ampli_biaisée(x,10) : est $\approx 10\%$-correct
  - ampli_biaisée(x,30) : est $\approx 25\%$-correct
  - ampli_biaisée(x,140) : est $\approx 75\%$-correct
  - ampli_biaisée(x,300) : est $\approx 95\%$-correct.

Cas biaisé : morale

Biased case: moral

true-biased or false-biased helps us

- vrai-biaisé ou faux-biaisé nous aide

  permet de conclure dès réponse "dans le sens du biais"

- permet de conclure dès réponse "dans le sens du biais"
- amplifier est possible quel que soit $p > 0$

  amplify is possible whatever $p > 0$

Unbiased case now

Cas non biaisé maintenant

Problème à réponse vrai/faux, algo p-correct

Problem with answer true / false, algo p-correct

Only option

Seule option :

repeat k times and take majority vote

- répéter k fois et prendre vote majoritaire
- voici la fonction (prendre k impair pour éviter ambiguité) :

here is the function (take k odd to avoid ambiguity):

```
fonction ampli_non_biaisée(x,k)
    V = 0
for pour i = 1 à k faire do
        if si A(x) alors V =+ 1
                      then
  if si V > k/2 alors
                then
        retourner vrai   return true
    sinon
else
        retourner faux   return false
```

Unbiased case

Cas non biaisé

(suite)
(cont.)

But, "reality check", let's consider : (coin flip)

- Mais, “reality check”, considérons :

not terrible

```
fonction pas_terrible
        si pile alors            if pile then
            retourner vrai       return true
        sinon                    else
            retourner faux       return false
```

This function

- Cette fonction
  - solves any true / false problem
  - résout n'importe quel problème à réponse vrai/faux
  - est $\frac{1}{2}$-correcte is 1/2-correct
  - est (bien sûr !) non biaisée
    is (of course!) unbiased
- Alors quoi ? ? So what ??

Cas non biaisé

(suite)
(cont.)

But, "reality check", let's consider : (coin flip)

- Mais, “reality check”, considérons :

not terrible

```
fonction pas_terrible
        si pile alors        if pile then
            retourner vrai   return true
        sinon                else
            retourner faux   return false
```

This function

- Cette fonction
  solves any true / false problem
  - résout n'importe quel problème à réponse vrai/faux
  - est $\frac{1}{2}$-correcte is 1/2-correct
  - est (bien sûr !) non biaisée
    is (of course!) unbiased
- Alors quoi ? ? So what ??
  Utopian to wait for a miracle when p-correct with p = 1
  - Utopique d'attendre un miracle lorsque p-correct avec $p = \frac{1}{2}$
  - Mais que peut-on espérer au juste ?
    But what can we hope for?

Unbiased case

Cas non biaisé

(suite)

(cont.)

Probability P[i, k] of i success among k attempts of a A (x) p-correct:

Probabilité $P[i,k]$ de i succès parmi k tentatives d'un $A(x)$ p-correct :

$$\binom{k}{i} p^i (1-p)^{k-i}$$

Probability that ampli_non_biaisé (x, k) is correct (k odd):

Probabilité que ampli_non_biaisé(x,k) soit correct (k impair) :

$$\sum_{i=\lceil \frac{k}{2} \rceil}^{k} P[i,k]$$

Unbiased case

Cas non biaisé

Confirmation : $p = \frac{1}{2}$ ne peut être amplifié

Confirmation: p = 1 can not be amplified

k	$\Pr[\text{ ampli\_non\_biaisé}(x,k) \text{ correct }] = \sum_{i=\lceil \frac{k}{2} \rceil}^{k} \binom{k}{i} p^i (1-p)^{k-i}$
1	$\frac{1}{2}$
3	$\left[\binom{3}{2} + \binom{3}{3}\right] \cdot \left(\frac{1}{2}\right)^3 =$

Unbiased case

Cas non biaisé

Confirmation : $p = \frac{1}{2}$ ne peut être amplifié

Confirmation: p = 1 can not be amplified

k	$\Pr[\text{ ampli\_non\_biaisé}(x,k) \text{ correct }] = \sum_{i=\lceil \frac{k}{2} \rceil}^{k} \binom{k}{i} p^i (1-p)^{k-i}$
1	$\frac{1}{2}$
3	$\left[\binom{3}{2} + \binom{3}{3}\right] \cdot (\frac{1}{2})^3 = [3+1] \cdot (\frac{1}{2})^3 = \frac{1}{2}$
5	$\left[\binom{5}{3} + \binom{5}{4} + \binom{5}{5}\right] \cdot (\frac{1}{2})^5 =$

Unbiased case

Cas non biaisé

Confirmation : $p = \frac{1}{2}$ ne peut être amplifié

Confirmation: p = 1 can not be amplified

k	Pr[ampli_non_biaisé(x, k) correct] $= \sum_{i=\lceil \frac{k}{2} \rceil}^{k} \binom{k}{i} p^i (1-p)^{k-i}$
1	$\frac{1}{2}$
3	$\left[\binom{3}{2} + \binom{3}{3}\right] \cdot (\frac{1}{2})^3 = [3+1] \cdot (\frac{1}{2})^3 = \frac{1}{2}$
5	$\left[\binom{5}{3} + \binom{5}{4} + \binom{5}{5}\right] \cdot (\frac{1}{2})^5 = [10+5+1] \cdot (\frac{1}{2})^5 = \frac{1}{2}$
$\vdots$	$\vdots$
$2m-1$	$\underbrace{\left[\binom{2m-1}{m} + \cdots + \binom{2m-1}{2m-1}\right]}_{\frac{1}{2} \cdot \left(\binom{2m-1}{0} + \binom{2m-1}{1} + \cdots + \binom{2m-1}{2m-1}\right)} \cdot (\frac{1}{2})^{2m-1} =$

Unbiased case

Cas non biaisé

Confirmation : $p = \frac{1}{2}$ ne peut être amplifié

Confirmation: p = 1 can not be amplified

k	$\Pr[\text{ ampli\_non\_biaisé}(x,k) \text{ correct }] = \sum_{i=\lceil \frac{k}{2} \rceil}^{k} \binom{k}{i} p^i (1-p)^{k-i}$
1	$\frac{1}{2}$
3	$\left[\binom{3}{2} + \binom{3}{3}\right] \cdot (\frac{1}{2})^3 = [3+1] \cdot (\frac{1}{2})^3 = \frac{1}{2}$
5	$\left[\binom{5}{3} + \binom{5}{4} + \binom{5}{5}\right] \cdot (\frac{1}{2})^5 = [10+5+1] \cdot (\frac{1}{2})^5 = \frac{1}{2}$
$\vdots$	$\vdots$
$2m-1$	$\underbrace{\left[\binom{2m-1}{m} + \cdots + \binom{2m-1}{2m-1}\right]}_{\frac{1}{2} \cdot \left(\binom{2m-1}{0} + \binom{2m-1}{1} + \cdots + \binom{2m-1}{2m-1}\right)} \cdot (\frac{1}{2})^{2m-1} = \frac{1}{2}$

Unbiased case

Cas non biaisé

Aucune amplification possible à moins que $p = \frac{1}{2} + \varepsilon > \frac{1}{2}$

No amplification possible unless p = 1/2 + ε > 1/2

Mais comment calculer un k raisonnable à l'aide de l'horrible

But how to calculate a reasonable k using the horrible

$$\sum_{i=\lceil \frac{k}{2} \rceil}^{k} \underbrace{\binom{k}{i} (\frac{1}{2} + \varepsilon)^i (\frac{1}{2} - \varepsilon)^{k-i}}_{P[i,k]} \quad ?$$

- À bras arms

Unbiased case

Cas non biaisé

Aucune amplification possible à moins que $p = \frac{1}{2} + \varepsilon > \frac{1}{2}$

No amplification possible unless p = 1/2 + ε > 1/2

Mais comment calculer un k raisonnable à l'aide de l'horrible

But how to calculate a reasonable k using the horrible

$$\sum_{i=\lceil \frac{k}{2} \rceil}^{k} \underbrace{\binom{k}{i}(\frac{1}{2}+\varepsilon)^i(\frac{1}{2}-\varepsilon)^{k-i}}_{P[i,k]} \quad ?$$

- À bras arms
  - $P[i, i+s+1] = P[i-1, i+s] \cdot (\frac{1}{2}+\varepsilon) + P[i, i+s] \cdot (\frac{1}{2}-\varepsilon)$
- À pied feets

Unbiased case

Cas non biaisé

Aucune amplification possible à moins que $p = \frac{1}{2} + \varepsilon > \frac{1}{2}$
No amplification possible unless p = 1/2 + ε > 1/2

Mais comment calculer un k raisonnable à l'aide de l'horrible
But how to calculate a reasonable k using the horrible

$$\sum_{i=\lceil \frac{k}{2} \rceil}^{k} \underbrace{\binom{k}{i} (\frac{1}{2} + \varepsilon)^i (\frac{1}{2} - \varepsilon)^{k-i}}_{P[i,k]} \quad ?$$

- À bras arms
  - $P[i, i+s+1] = P[i-1, i+s] \cdot (\frac{1}{2} + \varepsilon) + P[i, i+s] \cdot (\frac{1}{2} - \varepsilon)$
- À pied feets
  - par une formule du genre B&B problème 10.25
    by a formula like B & B problem 10.25
- À cheval horse

Unbiased case

Cas non biaisé

Aucune amplification possible à moins que $p = \frac{1}{2} + \varepsilon > \frac{1}{2}$

No amplification possible unless p = 1/2 + ε > 1/2

Mais comment calculer un k raisonnable à l'aide de l'horrible

But how to calculate a reasonable k using the horrible

$$\sum_{i=\lceil \frac{k}{2} \rceil}^{k} \underbrace{\binom{k}{i}(\frac{1}{2}+\varepsilon)^i(\frac{1}{2}-\varepsilon)^{k-i}}_{P[i,k]} \quad ?$$

- À bras arms
  - $P[i, i+s+1] = P[i-1, i+s] \cdot (\frac{1}{2}+\varepsilon) + P[i, i+s] \cdot (\frac{1}{2}-\varepsilon)$
- À pied feets
  - par une formule du genre B&B problème 10.25
    by a formula like B & B problem 10.25
- À cheval horse
  - par approximation statistique lorsque k est grand ($k \approx 30$)
    by statistical approximation when k is large (k ≈ 30)

Unbiased case

Cas non biaisé

Exemples : $p = \frac{1}{2} + \varepsilon$

To get

- Pour obtenir Pr[ampli_non_biaisé(x, k) correct] $=$ 95%
  - statistiques $\Longrightarrow k > 2,706\left(\frac{1}{4\varepsilon^2} - 1\right)$ OK
    statistics
  - take
    $\varepsilon = 5\% \Rightarrow$ prendre $k \approx 270$
  - take
    $\varepsilon = 1\% \Rightarrow$ prendre $k \approx 6750$
  - $\varepsilon = 0,5\% \Rightarrow$ prendre $k \approx 27000$
    take

To get

- Pour obtenir Pr[ampli_non_biaisé(x, k) correct] $=$ 99,5%
  - statistics
    statistiques $\Longrightarrow k > 6,636\left(\frac{1}{4\varepsilon^2} - 1\right)$ OK
  - pas tellement pire que pour 95%-correct.
    not so much worse than 95% -correct.

Cas non biaisé : morale
Unbiased case: moral

It is necessary (> 1/2) - correct starting

- Il faut ($> \frac{1}{2}$)-correct en partant
- Amplification lente Slow amplification
  - from 1%-correct to 95%-correct (biased)
    de 1%-correct à 95%-correct (biasé) : $k = 300$
  - de 51%-correct à 95%-correct (non biasé) : $k = 6750$
    from 51%-correct to 95%-correct (unbiased)
- Attention : ceci pour problèmes à réponses vrai/faux seulement
  Attention: this for problems with true / false answers only

Algorithmes de Las Vegas

B&B section 10.7

use the random to guide their choices

- utilisent l'aléat pour guider leurs choix
- ne se trompent jamais lorsqu'ils répondent

are never wrong when they respond

Las Vegas de type I : Las Vegas type I:

- répond toujours always answer
- mauvais choix ⇒ temps plus long wrong choice ⇒ longer time
  - sélection et médiane selection and median
  - quicksort quicksort
  - hashage (universal) hash

Las Vegas de type II : Las Vegas Type II:

- mauvais choix ⇒ l'algo déclare "pas capable"

  wrong choice ⇒ the algo declares "not able"
  - 8 reines 8 queens
  - factorisation entière

    whole factorization

Exemple : sélection et médiane

Example: selection and median

Reminder: selection of the kth element of a table

Rappel : sélection du $k$ième élément d'un tableau $T[1..n]$.

(seen) with pseudo-median as pivot, worst case time Θ (n)

- (vu) avec pseudo-médiane comme pivot, temps pire cas $\Theta(n)$
- (pas vu) pivot trivial $\Longrightarrow$ temps pire cas $\Theta(n^2)$
  (not seen) trivial pivot ⇒ worst case time Θ(n^2)
- (pas vu) pivot trivial $\Longrightarrow$ temps moyen $\Theta(n)$, constante cachée petite.
  (not seen) trivial pivot ⇒ average time Θ(n), small hidden constant.

Un algo Las Vegas de type I choisira le pivot au hasard...et alors ?

A Las Vegas type I algo will choose the pivot at random ... so what?

Fact: average time before = now expected time
Fait : temps moyen d'avant = temps espéré maintenant

- par la même preuve (pas vue)
  the same proof (not seen)
- en jouant de malchance, Las Vegas peut prendre autant de temps que le pire cas de l'algo à choix trivial
  playing bad luck, Las Vegas can take as much time as the worst case of the trivial choice algo
- peut même prendre ce pire temps sur un exemplaire qui aurait été facile pour l'algo à choix trivial !
  can even take this worst time on a copy that would have been easy for the trivial choice algo

But then, the interest?
Mais alors, l'intérêt ?

Selection and median

Sélection et médiane

(suite)
(cont.)

Fact: average time before = now expected time

Fait : temps moyen d'avant = temps espéré maintenant

- par la même preuve (pas vue)
  the same proof (not seen)
- en jouant de malchance, Las Vegas peut prendre autant de temps que le pire cas de l'algo à choix trivial
  playing bad luck, Las Vegas can take as much time as the worst case of the trivial choice algo
- peut même prendre ce pire temps sur un exemplaire qui aurait été facile pour l'algo à choix trivial !
  can even take this worst time on a copy that would have been easy for the trivial choice algo

But then, the interest?

Mais alors, l'intérêt ?

- There is no more bad copy!
  Il n'y a plus de mauvais exemplaire !
- l'algo prend aux riches et donne aux pauvres.
  the algo takes from the rich and gives to the poor.

Las Vegas de type I

Especially useful when a deterministic algo exists, which is:
Particulièrement utile quand un algo déterministe existe, qui est :

- bon en moyenne
  good on average
- mauvais en pire cas
  bad in worst case

So a Las Vegas will be able to:
Alors un Las Vegas pourra :

- éliminer les exemplaires pire cas eliminate the worst case examples
- uniformiser les exemplaires standardize copies
- maintenir un bon temps espéré maintain a good for expected time

Another example, quicksort (not seen)
Autre exemple, quicksort (pas vu) :

- $\Theta(n \log n)$ en moyenne on average
- quadratique en pire cas quadratic in the worst case
- devient temps espéré $\Theta(n \log n)$.
  becomes expected time

Las Vegas de type II
Las Vegas Type II

Reminder: such an algo can fail, but then detects its failure.
Rappel : un tel algo peut échouer, mais détecte alors son échec.

$$\texttt{fonction } \mathrm{LV}(x, y, \text{succès})$$

upon return:
- au retour :
  true success ⇒ there is solution of the copy x
  - succès vrai $\Longrightarrow$ y est solution de l'exemplaire x
  - succès faux $\Longrightarrow$ pas de chance false success ⇒ no luck
- $p(x)$ = probabilité de succès p (x) = probability of success
- $(\forall \text{ exemplaire } x)[p(x) > 0]$
  copy

Las Vegas de type II

Répétition non bornée d'un Las Vegas de type II

Unbounded Repeat of a Las Vegas Type II

stubborn(x)

```
fonction obstiné(x)
    répéter
        LV(x, y, succès)
    jusqu'à succès   until successful
    retourner y
```

always correct answer

- réponse toujours correcte
- toujours obtenue... always obtained ...
- ...un de ces jours !
  ...one of these days !

Las Vegas de type II

Mais quand obstiné(x) s'arrêtera-t-il ?
But when will stubborn (x) stop?

Soient Let

- p : probabilité de succès de LV p: probability of LV success
- s : temps espéré de LV en cas de succès s: expected time of LV if successful
- e : temps espéré de LV en cas d'échec e: expected time of LV in case of failure
- t : temps espéré de obstiné(x) t: expected time of obstinate (x)

Alors then

$$t = ps + (1 - p)(e + t)$$

d'où from where

$$t = s + \frac{1-p}{p} e$$

À noter : $s \downarrow$ ou $e \downarrow$ ou $p \uparrow \implies t \downarrow$
Note: s ↓ or e ↓ or p ↑ ⇒ t ↓

Las Vegas de type II

Exemple : les 8 reines

Example: the 8 queens

Fait expérimental : explorer le graphe des vecteurs ($k \leq 8$)-prometteurs par retour arrière examinait 114 sommets sur 2057 avant de trouver

Experimental fact: exploring the graph of the vectors (k ≤ 8)-prometers by backspace examined 114 vertices on 2057 before finding

Observation : les positions des reines qui résolvent le problème ont l'air plutôt arbitraires

Comment: Queen positions that solve the problem look rather arbitrary

Suggère un algo de Las Vegas : parmi les positions qui restent,

- choisir les positions successives à remplir au hasard
- abdiquer tout simplement si impasse atteinte

Suggest an algo from Las Vegas: among the remaining positions,
- choose successive positions to be filled randomly
- simply abdicate if deadlock reached

Las Vegas de type II pour les 8 reines
Las Vegas Type II for the 8 queens

Advantages
Avantages :

- conceptuellement plus simple que retour arrière
  conceptually simpler than backtracking

- faster in principle
  plus rapide en principe
  computer
  - $p = \Pr[\text{succès}] = 0{,}1293 = \frac{\#\text{ solutions}}{\#\text{ total}}$ (ordinateur)
  - s : temps espéré en cas de succès = coût de générer 9 vecteurs
  - e : temps espéré en cas d'échec = 6,971 vecteurs (ordinateur)
  - temps espéré de l'algo $= t = s + \frac{1-p}{p}e = 55,93$
  - ce 55,93 à comparer aux 114 par retour arrière !

- s: time expected on success = cost to generate 9 vectors
- e: expected time in case of failure = 6,971 vectors (computer)
- expected time of the algo = t = s + (1-p)/p*e = 55.93
- this 55.93 to compare to 114 by backtracking!

Las Vegas Type II for the 8 queens

Las Vegas de type II pour les 8 reines

(suite)
(cont.)

In practice: cost of generating random numbers cancels the gain in generated vectors

En pratique : coût de générer les nombres aléatoires annule le gain en vecteurs générés

Faire mieux ?
Oui, en ajustant s, e et p.

Do better ?
Yes, by adjusting s, e and p.

The idea: generate the first k queens randomly, and the last 8 - k by backtracking

L'idée : générer les k premières reines aléatoirement, et les $8 - k$ dernières par retour arrière :

three times faster than backtracking

- $k = 2$: trois fois plus rapide que retour arrière
- $k = 3$: seulement deux fois plus rapide, même si moins de vecteurs

only twice as fast, even if fewer vectors

Las Vegas type II for ... 39 queens

Las Vegas de type II pour les...39 reines

(suite)
(cont.)

L'avantage de Las Vegas sur le retour arrière croît lorsque n augmente
The advantage of Las Vegas on the backtracking increases when n increases

Example of n = 39:
Exemple de $n = 39$:

- par retour arrière : 10^{10} sommets avant la première solution
- pur Las Vegas : un million de fois plus rapide en implantation réelle
- hybride avec $k = 29$: deux millions de fois plus rapide en implantation, 20 millions moins de vecteurs

- by backspace: 1010 vertices before the first solution
- pure Las Vegas: a million times faster in real implementation
- hybrid with k = 29: two million times faster in implantation, 20 million fewer vectors

Exemple de $n = 1000$:
Example of n = 1000:

- bonne idée de choisir $k = 983$:-)
  good idea to choose k = 983 :-)

Las Vegas de type II

Exemple : factorisation entière

Las Vegas Type II
Example: whole factorization

- important problem
- no known effective algo (crypto relies on its difficulty!)
- does not seem yet NP-complete
- random choices + smart strategy + sophisticated estimates from number theory can sometimes succeed!

- problème important
- aucun algo efficace connu (la crypto repose sur sa difficulté !)
- ne semble pourtant pas NP-complet
- choix aléatoires + stratégie judicieuse + estimés sophistiqués tirés de la théorie des nombres permettent parfois de réussir !