

DUT SRC – IUT de Marne-la-Vallée
25/11/2011
INF120 - Algorithmique

Cours 4

Les tableaux et les boucles

Sources

- *Le livre de Java premier langage*, d'A. Tasso
- Cours INF120 de J.-G. Luque
- Cours de J. Henriet : <http://julienhenriet.olymp-network.com/Algo.html>

Plan du cours 4 – Tableaux et boucles

- Résumé des épisodes précédents
- Les tableaux
- Les boucles *while* et *for*

Plan du cours 4 – Tableaux et boucles

- Résumé des épisodes précédents
- Les tableaux
- Les boucles *while* et *for*

Résumé de l'épisode précédent

Méthodologie pour :

- **comprendre** un algorithme :
 - repérer les variables en **entrée**, en **sortie**, leur type,
 - choisir un exemple pour **faire la trace** de l'algorithme.

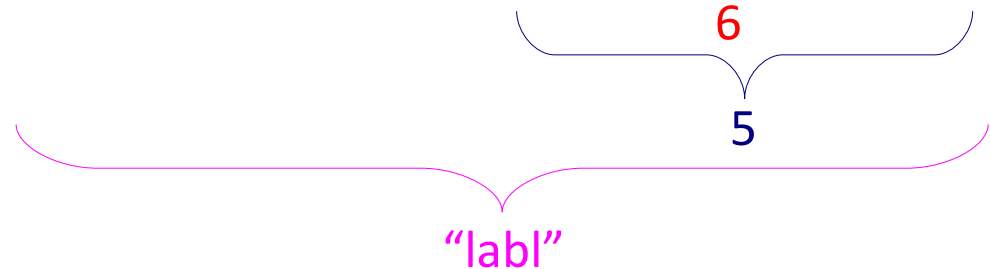
Résumé de l'épisode précédent

Méthodologie pour :

- **comprendre** un algorithme :

- repérer les variables en **entrée**, en **sortie**, leur type,
- choisir un exemple pour **faire la trace** de l'algorithme.

`SousChaine("blabla",2,Longueur("blabla")-1)`



Résumé de l'épisode précédent

Méthodologie pour :

- **comprendre** un algorithme :

- repérer les variables en **entrée**, en **sortie**, leur type,
- choisir un exemple pour **faire la trace** de l'algorithme.

- **écrire** un algorithme :

- repérer les variables en entrée, en sortie, leur type, choisir un exemple pour voir le **comportement attendu de l'algorithme sur un exemple**,
- trouver comment résoudre le problème **sur un exemple, étape par étape**,
- **généraliser** ces étapes en utilisant les **structures algorithmiques** à disposition (boucles, tests, appels d'autres algorithmes) et des **variables** pour stocker des valeurs qui varient à chaque étape.

Rappels sur la syntaxe (vocabulaire et façon de l'utiliser) Java et pseudo-code.

A garder en tête

DM à rendre pour le 12 décembre

Notes des TD2 et 3 à venir. Si pas satisfaits, rendez le TD4 même si vous n'êtes pas dans la liste.

Séance facultative d'exercices d'entraînement du 2 décembre.

Pourront y participer ceux qui ont travaillé sur **le DM1 ou l'exercice supplémentaire** au bas de la page <http://tinyurl.com/INF120-2011S1>.

Plan du cours 4 – Tableaux et boucles

- Résumé des épisodes précédents
- Les tableaux
- Les boucles *while* et *for*

Le nombre d'utilisations d'un mot dans un texte

- Écrivez un algorithme **TrouveMot** qui prend en entrée une chaîne de caractères *mot* et une chaîne de caractères *texte*, et compte le nombre de fois que *mot* apparaît dans *texte*.

Exemple : *mot* = "fuck"

texte = " Nobody's fuckin' screaming, Craig! Wake the fuck up! "



Algorithme **TrouveMot** :

Entrées : chaîne de caractères *mot*, chaîne de caractères *texte*

Sorties : nombre de fois (entier) que *mot* apparaît dans *texte*

Début

$i \leftarrow 1$

$compteur \leftarrow 0$

Tant que $i < \text{Longueur}(\text{texte}) - \text{Longueur}(\text{mot}) + 2$ faire :

 Si $\text{mot} = \text{SousChaîne}(\text{texte}, i, i + \text{Longueur}(\text{mot}) - 1)$ alors :

$compteur \leftarrow compteur + 1$

$i \leftarrow i + 1$

 FinSi

Fin TantQue

renvoyer *compteur*

Fin

Le nombre d'utilisations d'un mot dans un texte

La "minute culturelle"

Comment compter **plusieurs mots** dans un texte ?

Méthode naïve :

appliquer l'algorithme **TrouveMot** pour chaque mot à compter

➡ autant de lectures du texte que de mots distincts à compter

Le nombre d'utilisations d'un mot dans un texte

La "minute culturelle"

Comment compter **plusieurs mots** dans un texte ?

Méthode naïve :

appliquer l'algorithme **TrouveMot** pour chaque mot à compter

➡ autant de lectures du texte que de mots distincts à compter

Méthode astucieuse **plus rapide** :

faire une lecture du texte en construisant un dictionnaire

➡ compter tous les mots au fur et à mesure

➡ ensuite, pour avoir le nombre d'apparitions d'un mot,

simplement aller voir dans le dictionnaire

(pas besoin de reparcourir le texte)

Texte : "Nobody's fuckin' screaming, Craig! Wake the fuck up!"

Dictionnaire : N o b d y ' s f u c k i n r e a m g , C ! W t h p N o o b o o d d y y ' s s f u ...

↓
1 2 1 1 1 2 2 2 3 3 3 3 2 2 3 3 1 2 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 ...

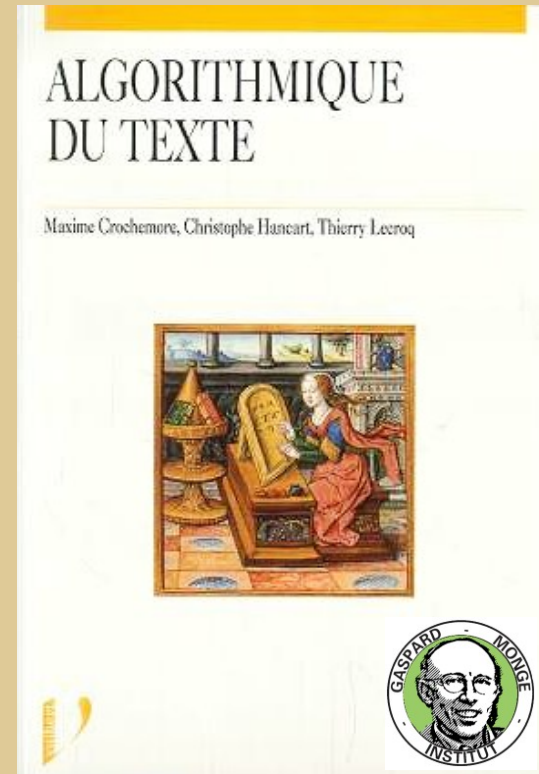
Le nombre d'utilisations d'un mot dans un texte

La “minute culturelle”

Comment compter **plusieurs mots** dans un texte ?

Problèmes d'“Algorithmique du texte”

➔ des experts au LIGM dans le
bâtiment Copernic !



<http://igm.univ-mlv.fr/~mac/CHL/CHL.html>

Les tableaux

Les tableaux sont des variables qui contiennent **plusieurs variables de même type**, stockées chacune dans une des cases du tableau.

Par exemple,

Un **tableau d'entiers** :

4
5
1
23
8
9

Un **tableau de chaînes de caractères** :

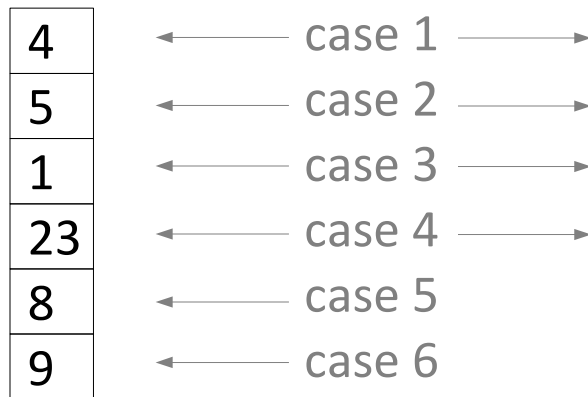
"chaine1"
"chaine2"
"blabla"
"toto"

Les tableaux

Les tableaux sont des variables qui contiennent **plusieurs variables de même type**, stockées chacune dans une des cases du tableau.

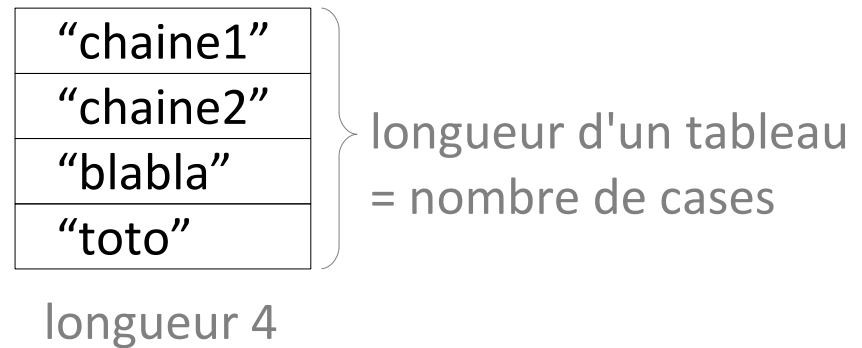
Par exemple,

Un **tableau d'entiers** :



longueur 6

Un **tableau de chaînes de caractères** :



Les tableaux

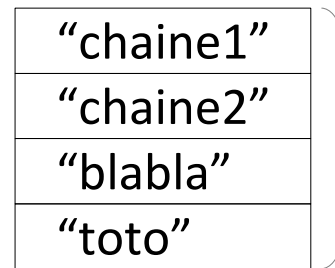
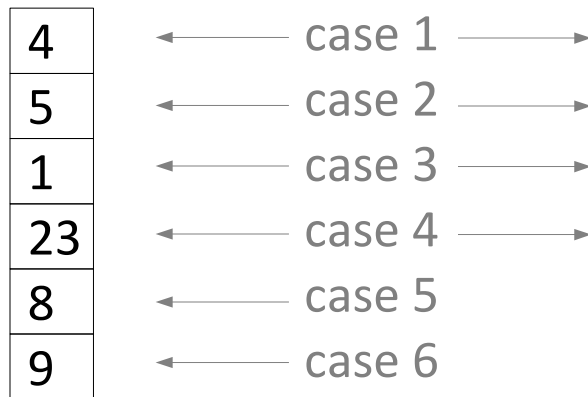
en pseudo-code

Les tableaux sont des variables qui contiennent **plusieurs variables de même type**, stockées chacune dans une des cases du tableau.

Par exemple, Variables : *tableau1*, un tableau d'entiers,
tableau2, un tableau de chaînes de caractères

Un tableau d'entiers :

Un tableau de chaînes de caractères :



longueur d'un tableau
= nombre de cases

Longueur(tableau2)

longueur 4

longueur 6

tableau1=NouveauTableau(6)

Case(tableau1,1)←4

Case(tableau1,2)←5

Case(tableau1,3)←1

Case(tableau1,4)←23

Case(tableau1,5)←8

Case(tableau1,6)←9

tableau2=NouveauTableau(4)

Case(tableau2,1)←"chaine1"

Case(tableau2,2)←"chaine2"

Case(tableau2,3)←"blabla"

Case(tableau2,4)←"toto"

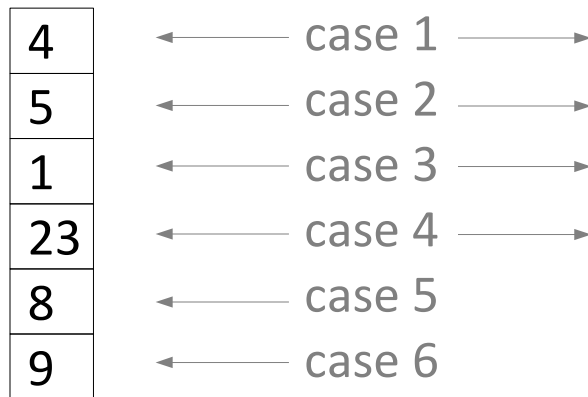
Les tableaux

en Java

Les tableaux sont des variables qui contiennent **plusieurs variables de même type**, stockées chacune dans une des cases du tableau.

Par exemple, `int[] tableau1; String[] tableau2;`

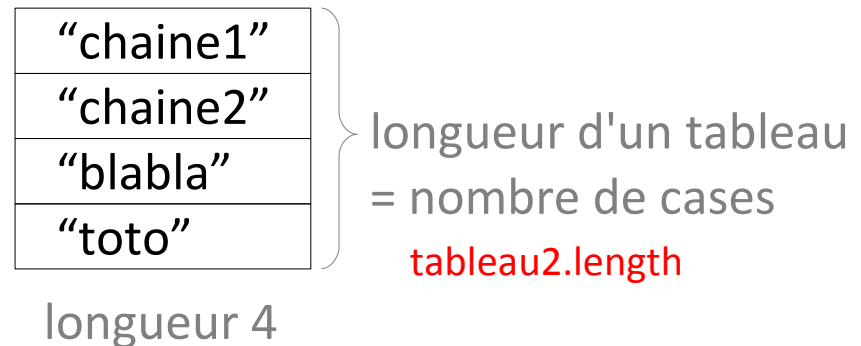
Un **tableau d'entiers** :



longueur 6

```
tableau1=new int[6];  
tableau1[0]=4;  
tableau1[1]=5;  
tableau1[2]=1;  
tableau1[3]=23;  
tableau1[4]=8;  
tableau1[5]=9;
```

Un **tableau de chaînes de caractères** :



```
tableau2=new String[4];  
tableau2[0]="chaine1";  
tableau2[1]="chaine2";  
tableau2[2]="blabla";  
tableau2[3]="toto";
```

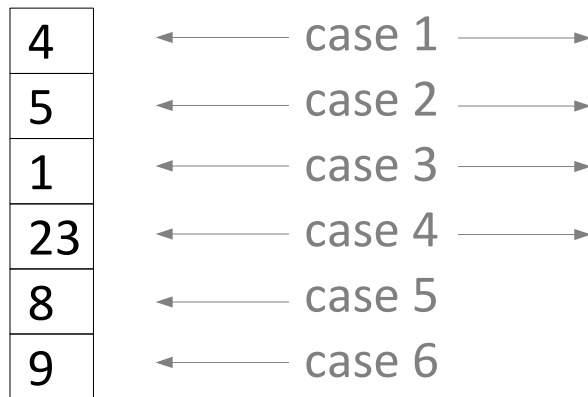
Attention, cases du tableau `t` numérotées de 0 à `t.length-1` en Java.

Les tableaux

Les tableaux sont des variables qui contiennent **plusieurs variables de même type**, stockées chacune dans une des cases du tableau.

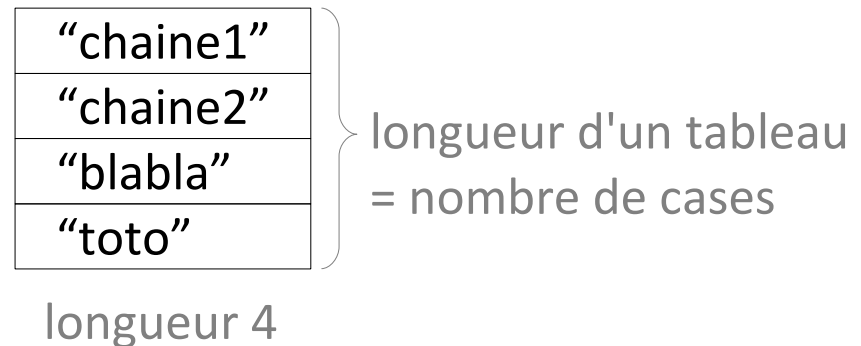
Par exemple,

Un **tableau d'entiers** :



longueur 6

Un **tableau de chaînes de caractères** :



```
//Afficher les cases du tableau tableau1
int i
i = 0
while (i<tableau1.length) {
    System.out.println(tableau1[i]);
    i = i+1
}
```

Les tableaux

Les tableaux sont des variables qui contiennent **plusieurs variables de même type**, stockées chacune dans une des cases du tableau.

Par exemple,

Un **tableau d'entiers** :

4
5
1
23
8
9

longueur 6

noté {4,5,1,23,8,9}

Un **tableau de chaînes de caractères** :

"chaine1"
"chaine2"
"blabla"
"toto"

longueur 4

noté {"chaine1","chaine2","blabla","toto"}

longueur d'un tableau
= nombre de cases

Les tableaux

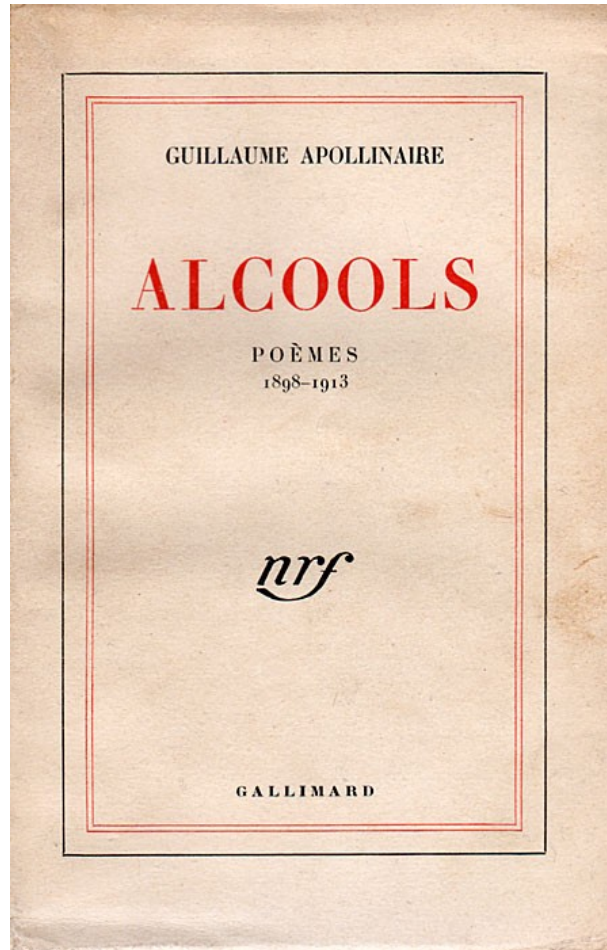
Pour lire le contenu d'un tableau...
il faut une **boucle pour aller lire chaque case** !

Si le tableau a été prévu trop court au début, **impossible de changer sa longueur**... il faut une boucle pour le recopier dans un tableau plus grand !

Possibilité de créer des **tableaux de tableaux**...

Manipulation et expériences en TD/TP...

Graphique du nombre d'apparitions des mots d'un texte



J'ai cueilli ce brin de bruyère
L'automne est morte souviens-t'en
Nous ne nous verrons plus sur terre
Odeur du temps brin de bruyère
Et souviens-toi que je t'attends

Graphique du nombre d'apparitions des mots d'un texte

Un tableau
de chaînes de
caractères *Mots*

j	1
ai	1
cueilli	1
ce	1
brin	2
de	2
bruyère	2
l	1
automne	1
est	1
morte	1
souviens	2
t	2
en	1
nous	2
ne	1
verrons	1
plus	1
sur	1
terre	1
odeur	1
du	1
temps	1
et	1
toi	1
que	1
je	1
attends	1

J'ai cueilli ce brin de bruyère
L'automne est morte souviens-t'en
Nous ne nous verrons plus sur terre
Odeur du temps brin de bruyère
Et souviens-toi que je t'attends

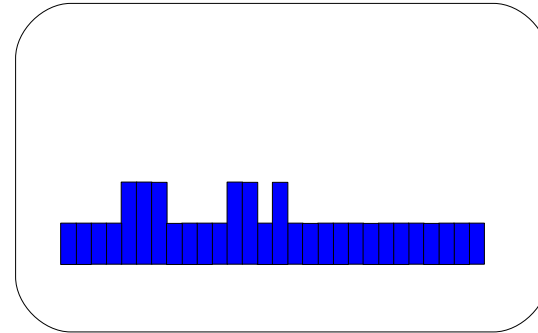
Un tableau d'entiers *NbApparitions*

Graphique du nombre d'apparitions des mots d'un texte

Un tableau
de chaînes de
caractères *Mots*

j	1
ai	1
cueilli	1
ce	1
brin	2
de	2
bruyère	2
l	1
automne	1
est	1
morte	1
souviens	2
t	2
en	1
nous	2
ne	1
verrons	1
plus	1
sur	1
terre	1
odeur	1
du	1
temps	1
et	1
toi	1
que	1
je	1
attends	1

Résultat voulu :

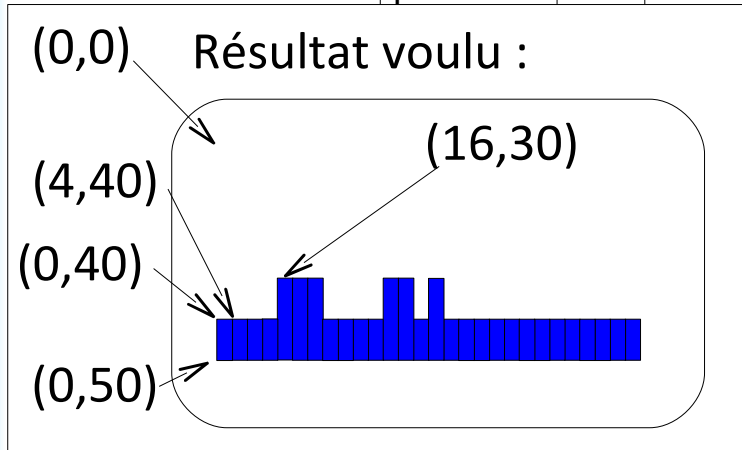


Un tableau d'entiers *NbApparitions*

Graphique du nombre d'apparitions des mots d'un texte

Un tableau
de chaînes de
caractères *Mots*

j	1
ai	1
cueilli	1
ce	1
brin	2
de	2
bruyère	2
l	1
automne	1
est	1
morte	1
souviens	2
t	2
en	1
nous	2
ne	1
verrons	1
plus	1



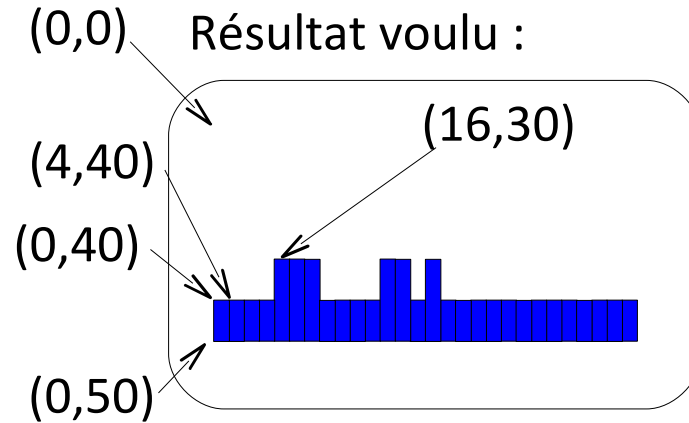
Un tableau d'entiers *NbApparitions*

Graphique du nombre d'apparitions des mots d'un texte

Un tableau
de chaînes de
caractères *Mots*

j
ai
cueilli
ce
brin
de
bruyère
l
automne
est
morte
souviens
t
en
nous
ne
verrons
plus
sur
terre
odeur
du
temps
et
toi
que
je
attends

1
1
1
1
2
2
2
1
1
1
1
2
2
1
2
1
1
1
1
1
1
1
1
1
1
1
1
1



Algorithme **DessineHistogramme**

Entrée : tableau de chaînes de caractères *Mots* et
tableau d'entiers *NbApparitions*.

Variable :

Début

...

```
dessineRectanglePlein(0,40,4,10,  
couleurRGB(0,0,255))
```

...

```
dessineRectanglePlein(16,30,4,20,  
couleurRGB(0,0,255))
```

...

Fin

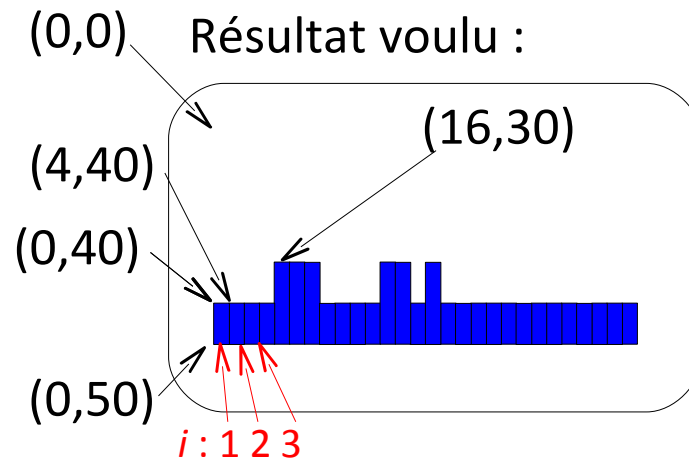
Un tableau d'entiers *NbApparitions*

Graphique du nombre d'apparitions des mots d'un texte

Un tableau
de chaînes de
caractères *Mots*

j
ai
cueilli
ce
brin
de
bruyère
l
automne
est
morte
souviens
t
en
nous
ne
verrons
plus
sur
terre
odeur
du
temps
et
toi
que
je
attends

1
1
1
1
2
2
2
1
1
1
2
2
1
2
1
1
1
1
1
1
1
1
1
1
1
1
1
1



Algorithme **DessineHistogramme**

Entrée : tableau de chaînes de caractères *Mots* et
tableau d'entiers *NbApparitions*.

Variable :

Début

...

x *y*

```
dessineRectanglePlein(0,40,4,10,  
couleurRGB(0,0,255))
```

h

...

```
dessineRectanglePlein(16,30,4,20,  
couleurRGB(0,0,255))
```

...

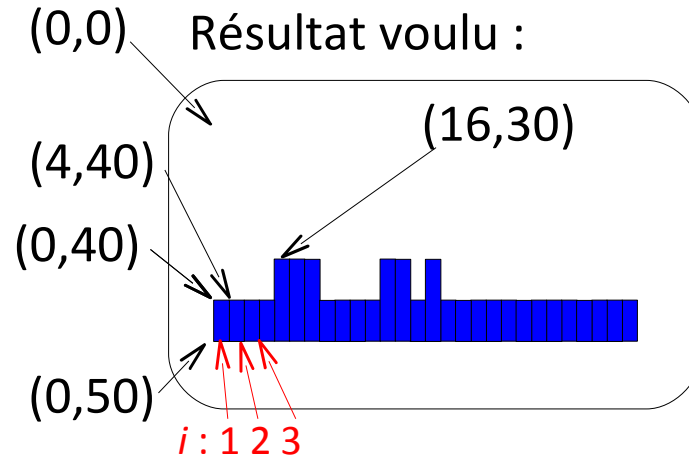
Fin

Un tableau d'entiers *NbApparitions*

Graphique du nombre d'apparitions des mots d'un texte

Un tableau
de chaînes de
caractères *Mots*

j	1
ai	1
cueilli	1
ce	1
brin	2
de	2
bruyère	2
l	1
automne	1
est	1
morte	1
souviens	2
t	2
en	1
nous	2
ne	1
verrons	1
plus	1
sur	1
terre	1
odeur	1
du	1
temps	1
et	1
toi	1
que	1
je	1
attends	1



Algorithme **DessineHistogramme**

Entrée : tableau de chaînes de caractères *Mots* et tableau d'entiers *NbApparitions*.

Variable :

Début

...

dessineRectanglePlein(0,40,4,10,
couleurRGB(0,0,255))

...

dessineRectanglePlein(16,30,4,20,
couleurRGB(0,0,255))

...

Fin

Un tableau d'entiers *NbApparitions*

$$x = 4(i-1)$$

$$y = 50 - 10 * \text{Case}(\text{NbApparitions}, i)$$

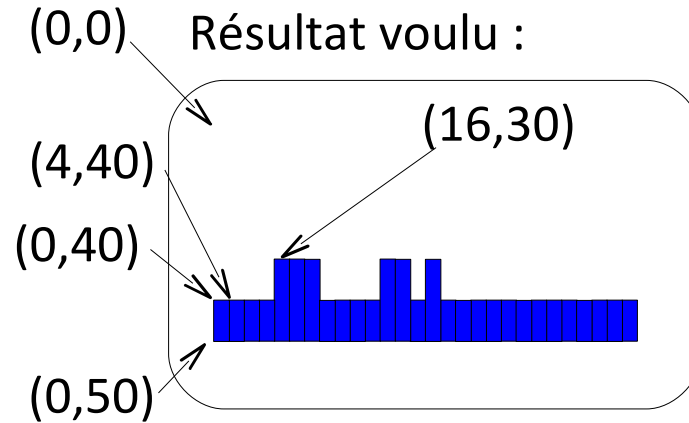
$$h = 10 * \text{Case}(\text{NbApparitions}, i)$$

Graphique du nombre d'apparitions des mots d'un texte

Un tableau
de chaînes de
caractères *Mots*

j
ai
cueilli
ce
brin
de
bruyère
l
automne
est
morte
souviens
t
en
nous
ne
verrons
plus
sur
terre
odeur
du
temps
et
toi
que
je
attends

1
1
1
1
2
2
2
1
1
1
1
2
2
1
2
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1



Algorithme **DessineHistogramme**

Entrée : tableau de chaînes de caractères *Mots* et
tableau d'entiers *NbApparitions*.

Variable : entier *compteur*

Début

$compteur \leftarrow 1$

Tant que ... faire :

dessineRectanglePlein($compteur * 4 - 4$,
 $50 - 10 * \mathbf{Case}(NbApparitions, compteur)$,
 $4, 10 * \mathbf{Case}(NbApparitions, compteur)$,
couleurRGB(0,0,255))

Fin TantQue

Fin

Un tableau d'entiers *NbApparitions*

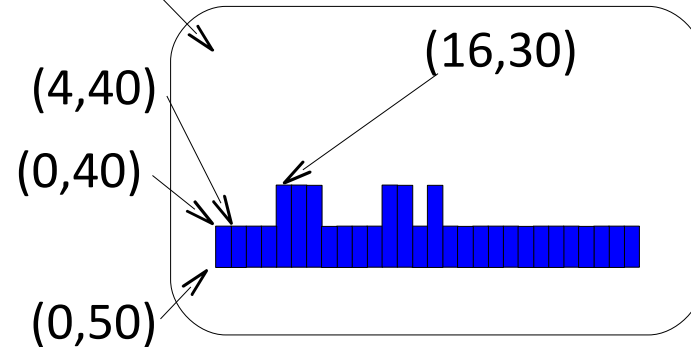
Graphique du nombre d'apparitions des mots d'un texte

Un tableau de chaînes de caractères *Mots*

j
ai
cueilli
ce
brin
de
bruyère
l
automne
est
morte
souviens
t
en
nous
ne
verrons
plus
sur
terre
odeur
du
temps
et
toi
que
je
attends

1
1
1
1
2
2
2
1
1
1
1
2
2
1
2
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1

(0,0) Résultat voulu :



Algorithme **DessineHistogramme**

Entrée : tableau de chaînes de caractères *Mots* et tableau d'entiers *NbApparitions*.

Variable : entier *compteur*

Début

compteur \leftarrow 1

Tant que *compteur* < Longueur(*Mots*)+1 faire :

dessineRectanglePlein(*compteur**4-4,
50-10***Case**(*NbApparitions*,*compteur*),

4,10***Case**(*NbApparitions*,*compteur*),

couleurRGB(0,0,255))

compteur \leftarrow 1 + *compteur*

Fin TantQue

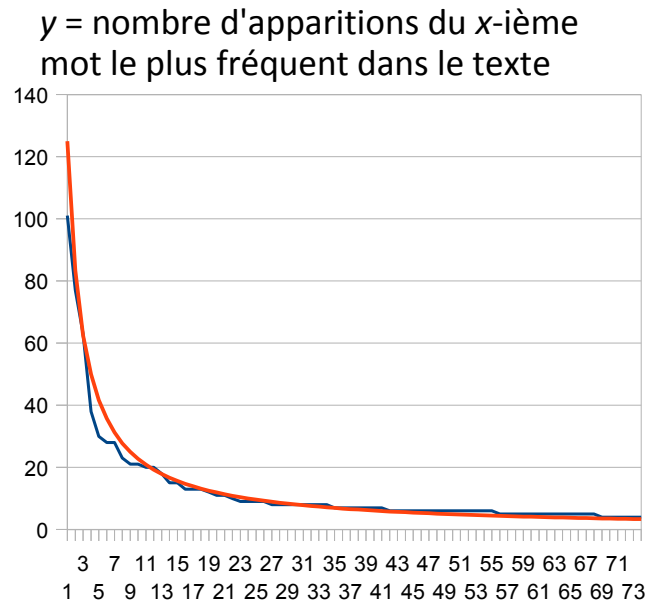
Fin

Un tableau d'entiers *NbApparitions*

Le nombre d'apparitions d'un mot dans un texte

La "minute mathématique"

La loi de Zipf prédit la courbe du nombre d'apparitions des mots les plus fréquents d'un texte.



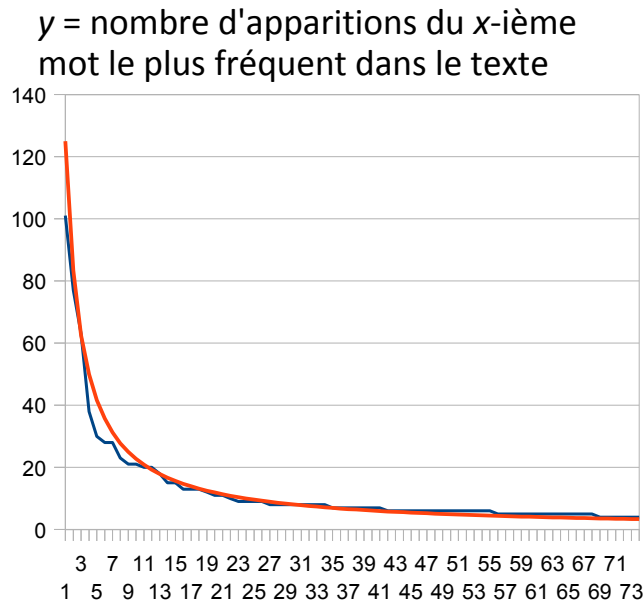
— Nombre réel d'apparition des mots
— Nombre estimé d'apparition des mots

x = numéro du mot (1 pour le plus fréquent, 2 pour le 2^o plus fréquent...)

Le nombre d'apparitions d'un mot dans un texte

La "minute mathématique"

La loi de Zipf prédit la courbe du nombre d'apparitions des mots les plus fréquents d'un texte.



— Nombre réel d'apparition des mots
— Nombre estimé d'apparition des mots

$$y=250/(x+1)$$

x = numéro du mot (1 pour le plus fréquent, 2 pour le 2° plus fréquent...)

Fonctionne pour n'importe quel texte assez long...

Plan du cours 4 – Tableaux et boucles

- Résumé des épisodes précédents
- Les tableaux
- Les boucles *while* et *for*

La boucle “for” / “Pour tout...”

La boucle “for” / “Pour tout”

Une boucle pour **parcourir tous les entiers entre deux valeurs entières.**

La boucle “for” / “Pour tout...”

La boucle “for” / “Pour tout”

Une boucle pour **parcourir tous les entiers entre deux valeurs entières.**

Algorithme **DessineHistogramme**

Entrée : tableau de chaînes de caractères *Mots* et tableau d'entiers *NbApparitions*.

Variable : entier *compteur*

Début

compteur ← 1

Tant que *compteur* < **Longueur**(*Mots*)+1 **faire** :

dessineRectanglePlein(*compteur**4-4,
 50-10***Case**(*NbApparitions*,*compteur*),
 4,10***Case**(*NbApparitions*,*compteur*),
 couleurRGB(0,0,255))

compteur ← 1 + *compteur*

Fin TantQue

Fin

Algorithme **DessineHistogramme**

Entrée : tableau de chaînes de caractères *Mots* et tableau d'entiers *NbApparitions*.

Variable : entier *compteur*

Début

Pour *compteur* de 1 à **Longueur**(*Mots*) **faire** :

dessineRectanglePlein(*compteur**4-4,
 50-10***Case**(*NbApparitions*,*compteur*),
 4,10***Case**(*NbApparitions*,*compteur*),
 couleurRGB(0,0,255))

compteur ← 1 + *compteur*

Fin Pour

Fin

La boucle “for” / “Pour tout...”

La boucle “for” / “Pour tout”

Une boucle pour **parcourir tous les entiers entre deux valeurs entières.**

Algorithme **DessineHistogramme**

Entrée : tableau de chaînes de caractères *Mots* et tableau d'entiers *NbApparitions*.

Variable : entier *compteur*

Début

compteur ← 1

Tant que *compteur* < **Longueur**(*Mots*)+1 **faire** :

dessineRectanglePlein(*compteur**4-4,
50-10***Case**(*NbApparitions*,*compteur*),
4,10***Case**(*NbApparitions*,*compteur*),
couleurRGB(0,0,255))

compteur ← 1 + *compteur*

Fin TantQue

Fin

En Java :

```
int compteur;  
compteur=1;  
while (compteur<mots.length+1) {  
    ...  
}
```

Algorithme **DessineHistogramme**

Entrée : tableau de chaînes de caractères *Mots* et tableau d'entiers *NbApparitions*.

Variable : entier *compteur*

Début

Pour *compteur* de 1 à **Longueur**(*Mots*) **faire** :

dessineRectanglePlein(*compteur**4-4,
50-10***Case**(*NbApparitions*,*compteur*),
4,10***Case**(*NbApparitions*,*compteur*),
couleurRGB(0,0,255))

compteur ← 1 + *compteur*

Fin Pour

Fin

```
for (int compteur=1;compteur<mots.length+1;compteur++) {  
    ...  
}
```

La boucle “for” / “Pour tout...”

La boucle “for” / “Pour tout”

Une boucle pour **parcourir tous les entiers entre deux valeurs entières.**

Algorithme **DessineHistogramme**

Entrée : tableau de chaînes de caractères *Mots* et tableau d'entiers *NbApparitions*.

Variable : entier *compteur*

Début

compteur ← 1

Tant que *compteur* < **Longueur**(*Mots*)+1 **faire** :

dessineRectanglePlein(*compteur**4-4,
 50-10***Case**(*NbApparitions*,*compteur*),
 4,10***Case**(*NbApparitions*,*compteur*),
 couleurRGB(0,0,255))

compteur ← 1 + *compteur*

Fin TantQue

Fin

En Java :

```
int compteur;  
compteur=1;  
while (compteur<mots.length+1) {  
    ...  
}
```

Algorithme **DessineHistogramme**

Entrée : tableau de chaînes de caractères *Mots* et tableau d'entiers *NbApparitions*.

Variable : entier *compteur*

Début

Pour *compteur* de 1 à **Longueur**(*Mots*) **faire** :

dessineRectanglePlein(*compteur**4-4,
 50-10***Case**(*NbApparitions*,*compteur*),
 4,10***Case**(*NbApparitions*,*compteur*),
 couleurRGB(0,0,255))

compteur ← 1 + *compteur*

Fin Pour

Fin

*déclaration +
initialisation*

```
for (int compteur=1; compteur<mots.length+1; compteur++) {  
    ...  
}
```

La boucle “for” / “Pour tout...”

La boucle “for” / “Pour tout”

Une boucle pour **parcourir tous les entiers entre deux valeurs entières.**

Algorithme **DessineHistogramme**

Entrée : tableau de chaînes de caractères *Mots* et tableau d'entiers *NbApparitions*.

Variable : entier *compteur*

Début

compteur ← 1

Tant que *compteur* < **Longueur**(*Mots*)+1 **faire :**

dessineRectanglePlein(*compteur**4-4,
 50-10***Case**(*NbApparitions*,*compteur*),
 4,10***Case**(*NbApparitions*,*compteur*),
 couleurRGB(0,0,255))

compteur ← 1 + *compteur*

Fin TantQue

Fin

En Java :

```
int compteur;  
compteur=1;  
while (compteur<mots.length+1) {  
    ...  
}
```

Algorithme **DessineHistogramme**

Entrée : tableau de chaînes de caractères *Mots* et tableau d'entiers *NbApparitions*.

Variable : entier *compteur*

Début

Pour *compteur* de 1 à **Longueur**(*Mots*) **faire :**

dessineRectanglePlein(*compteur**4-4,
 50-10***Case**(*NbApparitions*,*compteur*),
 4,10***Case**(*NbApparitions*,*compteur*),
 couleurRGB(0,0,255))

compteur ← 1 + *compteur*

Fin Pour

Fin

*déclaration +
initialisation* *condition d'arrêt*

```
for (int compteur=1; compteur<mots.length+1; compteur++) {  
    ...  
}
```

La boucle “for” / “Pour tout...”

La boucle “for” / “Pour tout”

Une boucle pour **parcourir tous les entiers entre deux valeurs entières.**

Algorithme **DessineHistogramme**

Entrée : tableau de chaînes de caractères *Mots* et tableau d'entiers *NbApparitions*.

Variable : entier *compteur*

Début

compteur ← 1

Tant que *compteur* < **Longueur**(*Mots*)+1 **faire** :

dessineRectanglePlein(*compteur**4-4,
 50-10***Case**(*NbApparitions*,*compteur*),
 4,10***Case**(*NbApparitions*,*compteur*),
 couleurRGB(0,0,255))

compteur ← 1 + *compteur*

Fin TantQue

Fin

En Java :

```
int compteur;  
compteur=1;  
while (compteur<mots.length+1) {  
    ...  
}
```

Algorithme **DessineHistogramme**

Entrée : tableau de chaînes de caractères *Mots* et tableau d'entiers *NbApparitions*.

Variable : entier *compteur*

Début

Pour *compteur* de 1 à **Longueur**(*Mots*) **faire** :

dessineRectanglePlein(*compteur**4-4,
 50-10***Case**(*NbApparitions*,*compteur*),
 4,10***Case**(*NbApparitions*,*compteur*),
 couleurRGB(0,0,255))

compteur ← 1 + *compteur*

Fin Pour

Fin

*déclaration +
initialisation* *condition d'arrêt* *mise à jour*

```
for (int compteur=1; compteur<mots.length+1; compteur++) {  
    ...  
}
```

La boucle “for” / “Pour tout...”

La boucle “for” / “Pour tout”

Une boucle pour **parcourir tous les entiers entre deux valeurs entières.**

Algorithme **DessineHistogramme**

Entrée : tableau de chaînes de caractères *Mots* et tableau d'entiers *NbApparitions*.

Variable : entier *compteur*

Début

compteur ← 1

Tant que *compteur* < **Longueur**(*Mots*)+1 **faire :**

dessineRectanglePlein(*compteur**4-4,
50-10***Case**(*NbApparitions*,*compteur*),
4,10***Case**(*NbApparitions*,*compteur*),
couleurRGB(0,0,255))

compteur ← 1 + *compteur*

Fin TantQue

Fin

En Java :

```
int compteur;  
compteur=1;  
while (compteur<mots.length+1) {  
    ...  
}
```

Algorithme **DessineHistogramme**

Entrée : tableau de chaînes de caractères *Mots* et tableau d'entiers *NbApparitions*.

Variable : entier *compteur*

Début

Pour *compteur* de 1 à **Longueur**(*Mots*) **faire :**

dessineRectanglePlein(*compteur**4-4,
50-10***Case**(*NbApparitions*,*compteur*),
4,10***Case**(*NbApparitions*,*compteur*),
couleurRGB(0,0,255))

compteur ← 1 + *compteur*

Fin Pour

Fin

déclaration + initialisation *condition d'arrêt* *mise à jour*

```
for (int compteur=1; compteur<mots.length+1; compteur++) {  
    ...  
    compteur=compteur+1  
}
```

The diagram illustrates the execution of the for loop. It shows the declaration and initialization of the counter variable 'compteur' to 1, the condition 'compteur < mots.length + 1', and the update 'compteur++'. Arrows indicate the flow of the loop, showing the counter being incremented by 1 in each iteration.

Comptage des mots d'un texte

Prétraitement du texte : remplacer ' et – par un espace.

entrée :

j'ai cueilli ce brin de bruyère
l'automne est morte souviens-t'en
nous ne nous verrons plus sur terre
odeur du temps brin de bruyère
et souviens-toi que je t'attends

sortie :

j ai cueilli ce brin de bruyère
l automne est morte souviens t en
nous ne nous verrons plus sur terre
odeur du temps brin de bruyère
et souviens toi que je t attends

Comptage des mots d'un texte

Prétraitement du texte : remplacer ' et – par un espace.

Algorithme **RemplacePonctuation**

Entrée : chaîne de caractères *texte*

Sortie : chaîne de caractères

Variable : entier *i*

Début

$i \leftarrow 1$

Pour tout *i* de 1 à **Longueur**(*texte*) faire :

 Si **Caractère**(*texte*,*i*)="'" OU **Caractère**(*texte*,*i*)="–" alors :

texte \leftarrow ...

 FinSi

$i \leftarrow i + 1$

FinPour

renvoyer *texte*

Fin

entrée :

j'ai cueilli ce brin de bruyère
l'automne est morte souviens-t'en
nous ne nous verrons plus sur terre
odeur du temps brin de bruyère
et souviens-toi que je t'attends

sortie :

j ai cueilli ce brin de bruyère
l automne est morte souviens t en
nous ne nous verrons plus sur terre
odeur du temps brin de bruyère
et souviens toi que je t attends

Comptage des mots d'un texte

Prétraitement du texte : remplacer ' et – par un espace.

Algorithme **RemplacePonctuation**

Entrée : chaîne de caractères *texte*

Sortie : chaîne de caractères

Variable : entier *i*

Début

$i \leftarrow 1$

Pour tout *i* de 1 à **Longueur**(*texte*) faire :

Si **Caractère**(*texte*,*i*)="'" OU **Caractère**(*texte*,*i*)="-" alors :

texte \leftarrow **SousChaine**(*texte*,1,*i*-1) " " **SousChaine**(*texte*,*i*+1,**Longueur**(*texte*))

 FinSi

$i \leftarrow i + 1$

FinPour

renvoyer *texte*

Fin

entrée :

j'ai cueilli ce brin de bruyère
l'automne est morte souviens-t'en
nous ne nous verrons plus sur terre
odeur du temps brin de bruyère
et souviens-toi que je t'attends

sortie :

j ai cueilli ce brin de bruyère
l automne est morte souviens t en
nous ne nous verrons plus sur terre
odeur du temps brin de bruyère
et souviens toi que je t attends