



Introduction aux séquences intermédiaires

Savoir raisonner sur une séquence de plus haut niveau que la séquence fournie

Jean-Michel Adam - UGA - UFR SHS

Notion de séquence intermédiaire

- La résolution d'un problème se fait en général par la combinaison de parcours et de recherches.
- Dans certaines situations il est difficile de trouver directement une solution en raisonnant à partir de la séquence fournie.
- Le fait de raisonner sur une **séquence de plus haut niveau** permet de concevoir d'autres solutions.
- Ces solutions sont basées sur une autre vision du problème posé.

Exemple de séquence intermédiaire

- La séquence donnée est un texte formé de lettres et d'espaces
- La séquence donnée peut être interprétée
 - comme une séquence de caractères,
 - comme une séquence de doublets,
 - comme une séquence de mots,
 - etc.

Exemple de séquence intermédiaire

- Une séquence de durées représentée sur un fichier
- Pour calculer la durée moyenne de la séquence, il est plus facile de raisonner sur la séquences des durées converties en secondes, c'est-à-dire sur une séquence d'entiers

Exercice

- On considère un texte formé de lettres et d'espaces représenté dans un fichier.
- Un mot est une suite de lettres.
- Le fichier peut commencer par un (des) espace(s) ou par la première lettre du premier mot.
- Le dernier mot peut être suivi d'un (de plusieurs) espace(s) avant la marque de fin de fichier.
- Ecrire un algorithme qui calcule le **nombre de mots** du texte.

Comment compter les mots ?

Exemple:

_cette ___ phrase _contient ___ sept __ mots __ █

- On peut compter
 - les débuts de mots : espaces suivis d'une lettre
 - les fins de mots : lettres suivies d'un espace
- Pour cela on raisonne sur une **séquence de doublets de caractères** afin de repérer les débuts ou fins de mots:

_ c ce et tt te e _ _ _ _ p ph hr ra as se e _ ...

Si on compte les débuts de mots, comment faire si le texte commence par une lettre ?

Définition de la séquence intermédiaire de doublets

Le doublet courant est formé du caractère courant et du caractère qui le précède

Lexique

f : fichier de caractères // donnée : texte examiné

cp : caractère // inter: dernier caractère de la partie parcourue de f

EC	cp, f.ec
FDS	f.fdf
AV	cp ← f.ec f.lireSuivant
DEM	cp ← ' ' f.lirePremier

Compter les mots du texte revient à compter les doublets caractérisant un début ou une fin de mot

Compter les mots d'un texte

Lexique principal

f : fichier de caractères // texte examiné
cp : caractère // inter: dernier caractère de la partie parcourue
e : écran // périphérique de sortie
nbm : entier ≥ 0 // nombre de débuts de mots de la pp de f

Algorithme principal

// on compte les débuts de mots

cp \leftarrow ' ' ; f.lirePremier

nbm \leftarrow 0

tantque non f.fdf faire

si cp = ' ' et f.ec \neq ' '

alors

nbm \leftarrow nbm + 1

fsi

cp \leftarrow f.ec ; f.lireSuivant

ftq

e.afficher("nombre de mots : ", nbm)

Quelles modifications apporter
si l'on veut compter les fins de mots ?

Compter les mots d'un texte (fins de mot)

Lexique principal

f : fichier de caractères // texte examiné

cp : caractère // inter: dernier caractère de la partie parcourue

e : écran // périphérique de sortie

nbm : entier ≥ 0 // nombre de fins de mots rencontrés

Algorithme principal

// on compte les fins de mots

cp \leftarrow ' ' ; f.lirePremier // DEM doublet

nbm \leftarrow 0

tantque non f.fdf faire

si cp \neq ' ' et f.ec = ' ' // on compte les fins de mots

alors

nbm \leftarrow nbm + 1

fsi

cp \leftarrow f.ec; f.lireSuivant // AV doublet

ftq

// le dernier caractère du dernier mot peut être suivi de la marque de fin de fichier

si cp \neq ' ' alors nbm \leftarrow nbm + 1 fsi

e.afficher("nombre de mots : ", nbm)

Calculer la longueur moyenne des mots d'un texte

Exemple :

_cette ___ phrase _contient ___ sept __ mots __ █

- On peut raisonner

- sur des doublets

c ce et tt te e __ __ _p ph hr ra as se e_ ...

il suffit de compter :

- les débuts de mot
- les lettres

Calculer la longueur moyenne des mots d'un texte

Lexique principal

f : fichier de caractères // texte examiné
cp : caractère // inter: dernier caractère de la partie parcourue
e : écran // périphérique de sortie
nbm : entier ≥ 0 // nombre de débuts de mots de la pp de f
nbl : entier ≥ 0 // nombre de lettres de la pp de f

Algorithme principal

// on compte les débuts de mots

cp \leftarrow ' ' ; f.lirePremier

nbm \leftarrow 0 ; nbl \leftarrow 0

tantque non f.fdf faire

 si f.ec \neq ' ' alors nbl \leftarrow nbl + 1 fsi

si cp = ' ' et f.ec \neq ' '

alors

 nbm \leftarrow nbm + 1

fsi

 cp \leftarrow f.ec ; f.lireSuivant

ftq

e.afficher("nombre de mots : ", nbm)

si nbm \neq 0 alors e.afficher("longueur moyenne des mots : " , nbl/nbm)

fsi

Calculer la longueur moyenne des mots d'un texte

Exemple :

_cette ___ phrase _contient ___ sept __ mots __ █

- On peut raisonner

- sur des doublets
- sur la séquence des longueurs des mots du texte :

_cette ___ phrase _contient ___ sept __ mots __ █
5 6 8 4 4

il suffit de calculer la moyenne de la séquence d'entiers

Construction de la séquence intermédiaire des longueurs de mots

Lexique

f : fichier de caractères // texte examiné

lmc : entier ≥ 0 // intermédiaire : longueur du mot courant

- Pour construire la séquence intermédiaire de longueurs mots, il faut définir:
 - EC
 - FDS
 - AV
 - DEM

Réalisation de la séquence intermédiaire de longueurs de mots

EC	lmc
FDS	lmc = 0
AV	<p><u>tantque</u> ¬ f.fdf <u>etpuis</u> f.ec = ' ' <u>faire</u> // ignorer les espaces qui précèdent le mot f.lireSuivant</p> <p><u>ftq</u> // f.fdf oualors f.ec ≠ ' '</p> <p>lmc ← 0</p> <p><u>tantque</u> ¬ f.fdf <u>etpuis</u> f.ec ≠ ' ' <u>faire</u> // parcours du mot lmc ← lmc + 1 f.lireSuivant</p> <p><u>ftq</u> // f.fdf oualors f.ec = ' ', lmc = longueur du mot parcouru ou 0 si pas plus de mot</p>
DEM	f.lirePremier AV

À transformer
en action
nommée
avancerLgMot

Réalisation de l'action avancerLgMot

action avancerLgMot (modifié f:fichier de caractère, élaboré lg : entier ≥ 0)

// Effet: lit le mot suivant s'il existe et calcule sa longueur

// E.I. lg = ? , f.fdf oualors f.ec = caractère qui suit le dernier mot parcouru

// E.f. : lg = longueur du mot parcouru ou 0, f.fdf oualors f.ec = ' '

Lexique de avancerLgMot

f: fichier de caractères // paramètre : séquence à parcourir

lg : entier ≥ 0 // paramètre : longueur de la pp du prochain mot ou 0 si pas de mot

Algorithme de avancerLgMot

tantque \neg f.fdf etpuis f.ec = ' ' faire

f.lireSuivant

ftq

// f.fdf oualors f.ec \neq ' '

lg \leftarrow 0

tantque \neg f.fdf etpuis f.ec \neq ' ' faire

lg \leftarrow lg + 1

f.lireSuivant

ftq

// f.fdf oualors f.ec = ' ', si lg = 0 alors on est sur la fin de la séquence de longueurs de mots

Longueur moyenne des mots d'un texte

Lexique principal

f : fichier de caractères // texte examiné
e : écran // périphérique de sortie
lmc : entier ≥ 0 // longueur du mot courant
nbm : entier ≥ 0 // nombre de mots rencontrés
s : entier ≥ 0 // somme des longueurs des mots rencontrés
Action utilisée : avancerLgMot

Algorithme principal

f.lirePremier

avancerLgMot(f,lmc)

s \leftarrow 0 ; nbm \leftarrow 0

tantque lmc \neq 0 faire

 s \leftarrow s + lmc ; nbm \leftarrow nbm + 1

 avancerLgMot(f,lmc)

ftq

// lmc = 0 \Rightarrow on a parcouru tous les mots

si nbm = 0

alors e.afficher("aucun mot")

sinon e.afficher("longueur moyenne des mots du texte = " , s / nbm)

fsi

Construction d'une séquence intermédiaire de mots

- Si la valeur du mot est importante, il est intéressant de pouvoir énumérer une séquence de mots

Lexique

f : fichier de caractères // texte examiné

motc : chaîne // intermédiaire : mot courant

- Construire la séquence intermédiaire de mots
- Il faut définir:
 - EC
 - FDS
 - AV
 - DEM

Réalisation de la séquence intermédiaire de mots

EC	motc
FDS	motc = ""
AV	<p><u>tantque</u> \neg f.fdf <u>etpuis</u> f.ec = ' ' <u>faire</u> f.lireSuivant</p> <p><u>ftq</u> // f.fdf oualors f.ec \neq ' ' motc \leftarrow ""</p> <p><u>tantque</u> \neg f.fdf <u>etpuis</u> f.ec \neq ' ' <u>faire</u> motc \leftarrow motc • f.ec f.lireSuivant</p> <p><u>ftq</u> // f.fdf oualors f.ec = ' '</p>
DEM	f.lirePremier AV

À transformer
en action
nommée
avancerMot

Réalisation de l'action avancerMot

action avancerMot (modifié f: fichier de caractère, élaboré mot : chaine)

// Effet: lit le mot suivant s'il existe le copie dans la chaine mot

// E.I. mot = ?, f.fdf oualors f.ec = caractère qui suit le dernier mot parcouru

// E.f. : mot = mot parcouru ou "" si pas de mot, f.fdf oualors f.ec=' '

Lexique de avancerMot

f : fichier de caractères // paramètre : séquence à parcourir

mot : chaine // paramètre : pp du prochain mot

Algorithme de avancerMot

tantque \neg f.fdf etpuis f.ec = ' ' faire

f.lireSuivant

ftq

// f.fdf oualors f.ec \neq ' '

mot \leftarrow ""

tantque \neg f.fdf etpuis f.ec \neq ' ' faire

mot \leftarrow mot • f.ec

f.lireSuivant

ftq

// f.fdf oualors f.ec = ' '

// f.fdf oualors f.ec = ' ', si motc = "" alors on est sur la fin de la séquence de mots

