

DS - Algorithmes et Structures de Données

Mardi 10 janvier 2023

Durée 3h – Documents NON autorisés

1. Quand le récursif est plus simple que l'itératif – 5 pts

On dispose d'une liste de mots représentée par la structure de données suivante :

```
Type cellule : Enregistrement
    mot : chaîne
    suiv : ^cellule
    FinEnregistrement
liste : ^cellule
```

1.1. Ecrire en pseudo-langage une procédure **récursive** qui écrit les mots de la liste dans l'**ordre inverse**. Expliquer le principe récursif de cette procédure.

1.2. Ecrire en pseudo-langage une procédure **itérative** qui écrit les mots de la liste dans l'**ordre inverse**. Expliquer votre analyse.

2. Chromosomes – 5 pts

Les chromosomes sont constitués d'ADN qui porte les gènes d'un être vivant. On peut schématiser un chromosome comme 2 brins chacun constitué d'information sous la forme d'une série de lettres (A, C, T et G). On s'intéresse ici à deux opérations qui peuvent s'effectuer sur un brin de chromosome :

- la mutation : une lettre au sein d'un brin de chromosome est substituée à une autre.
- l'enjambement : deux brins de 2 chromosomes différents s'échangent une partie de leur chaîne pour donner 2 nouveaux brins.

Exemple :

b1 = TACTGCCTAGTCGGCGTTCGTTTCGCCTTAA

b2 = TGCGAATCCGTCACGTTGCATCAGGTCCAT

Mutation de b1 (en 12) : TACTGCCTAGT**A**GGCGTTCGTTTCGCCTTAA

Enjambement de b1 et b2 (en 9) : donne 2 nouveaux brins c1 et c2

c1 = TACTGCCT**CGTCACGTTGCATCAGGTCCAT**

c2 = TGCGAATC**AGTCGGCGTTCGTTTCGCCTTAA**

On se propose de représenter un brin par les 2 structures de données suivantes :

- Représentation par tableau
Type brin : tableau[1..30] de car
- Représentation par liste chaînée
Type code : Enregistrement
info : car
suiv : brin
FinEnregistrement
brin : ^code

2.1. Pour chacune des structures de données, écrire en pseudo-langage une procédure `faire_mutation` qui réalise une mutation sur un brin : l'emplacement et la substitution de la mutation seront choisis au hasard (`random(n)` tire un entier au hasard entre 0 et n-1).

Procédure `faire_mutation` (E/S b : brin)

2.2. Pour chacune des structures de données, écrire en pseudo-langage une procédure `faire_enjambement` qui réalise l'échange des parties de brin : le point où se fait l'échange sera tiré au hasard.

Procédure `faire_enjambement` (E b1, b2 : brin, S c1, c2 : brin)

3. Mots croisés – 10 pts

E	X	A	M	E	N
P	■	N	E	■	I
R	E	G	L	E	■
E	C	L	A	T	E
U	R	E	■	U	T
V	A	■	O	D	E
E	N	T	R	E	■

Const m = 7
n = 6

Type grille : tableau[1..m, 1..n] de car

Les cases noires seront représentées par le caractère espace ' '.

3.1. Ecrire en pseudo-langage une fonction `noirsuiv` qui cherche la case noire suivante dans la grille `g`, en partant de la case de coordonnées (i, j) , dans la direction spécifiée par le booléen `horiz` (si `horiz=vrai`, on cherche vers la droite, sinon on cherche vers le bas). La réponse est un numéro de colonne si `horiz=vrai`, un numéro de ligne si `horiz=faux` et 0 s'il n'y a pas de case noire dans la direction spécifiée.

Fonction `noirsuiv` (`g` : grille, `i, j` : entier, `horiz` : booléen) : entier

Exemple : `noirsuiv(g, 2, 2, vrai)` retourne 5
`noirsuiv(g, 2, 2, faux)` retourne 0

3.2. Seules sont considérées comme mots les suites de 2 lettres ou plus, horizontales ou verticales, (dans l'exemple, les mots sont : examen, ne, règle, éclate, ure, ut, va, ode, entre, épreuve, écran, angle, méla, or, étude, ni, été). Ecrire en pseudo-langage une fonction qui renvoie le nombre de mots de la grille.

3.3. On veut installer les mots dans un tableau de listes : chaque liste contient des mots de même longueur et est **triée dans l'ordre alphabétique**. Pour cela, on définit le type `tabliste` de la façon suivante :

```
Type cellule = Enregistrement
    mot : chaine
    suiv : liste
    FinEnregistrement
liste = ^cellule
tabliste = tableau[2..m] de liste
```

Faire le dessin correspondant à la grille donnée en exemple.

3.4. Ecrire en pseudo-langage une fonction qui remplit cette structure de données à partir de la grille de mots croisés.

3.5. On veut savoir si les mots sont tous différents. Ecrire en pseudo-langage une fonction qui renvoie un booléen selon si les mots de la grille sont tous distincts ou non. Cette fonction utilise la structure de données remplie au 3.4 et sera **optimisée**.