UNIVERSITÉ DE MONTPELLIER
UNIVERSITÉ DE MONTPELLIER - FACULTÉ DES SCIENCES ET TECHNIQUES
PLACE EUGÈNE BATAILLON - 34095 MONTPELLIER CEDEX 5

Licence 3 informatique – 2015/2016

HLIN608 – Examen n° 1

Algorithmique du texte

Annie CHATEAU & Sylvain DAUDÉ & Alban MANCHERON

Avant toute chose, il convient de rappeler certaines règles et bons usages relatifs aux épreuves. Ensuite, il est bien entendu que les copies, téléphones, messages des voisins et autres collègues ne font pas partie de l'ensemble des documents autorisés (vos notes de cours, de TD et de TP le sont). De surcroît, seul l'enseignant est habilité à répondre à vos questions pendant l'épreuve. Il vous est par ailleurs conseillé de lire attentivement le sujet dans son intégralité.

Pour les questions 4, 7 et 9, vous pouvez répondre soit sur le sujet (le cas échéant, reportez votre numéro d'anonymat dans le cadre ci-dessous), soit reporter les tableaux sur votre copie. Toutes les autres réponses doivent être reportées sur la copie d'examen.

Numéro d'anonymat :

Exercice 1 : Assemblage et graphes de k-mers (rouges)

Question 1 : On rappelle la définition d'un graphe de DE BRUIJN sur un alphabet $\mathcal{A}$ de taille n pour des mots de taille k. Il s'agit d'un graphe orienté $G = (V, A)$, tel que les sommets de V sont tous les mots de longueur k sur l'alphabet $\mathcal{A}$ et il y a un arc entre deux sommets s'il y a un chevauchement de taille $k-1$ entre ces sommets, c'est-à-dire si le suffixe de taille $k-1$ du sommet origine est également le préfixe de taille $k-1$ du sommet destination.

a) Combien y a-t-il de sommets dans ce graphe ?

b) Pour un mot représentant un chevauchement de taille $k-1$ donné, combien y a-t-il de sommets origines possibles ? De sommets destination ? Et donc, d'arcs qui sont étiquetés par ce mot ?

Question 2 : Soit $\mathcal{M} = \{catacat, gataca, catacaga\}$ un ensemble de mots, on veut représenter son graphe des k-mers pour $k = 3$.

a) Combien y a-t-il de $k-$mers dans un mot de longueur ℓ ?

b) Représentez l'ensemble des 3-mers des mots de $\mathcal{M}$ dans une structure de données d'indexation de votre choix parmi celles vues en cours.

c) Quelle est la complexité en temps et en mémoire de la construction de cette structure en fonction de la taille des mots de $\mathcal{M}$ et de k ?

d) Pour construire le graphe des $k-$mers à partir de cette structure, on s'intéresse à la requête suivante : est-ce que le suffixe de taille $k-1$ d'un élément de la structure est un préfixe de taille $k-1$ d'un autre élément. Quelle est la complexité de cette requête ?

e) Construisez le graphe de DE BRUIJN de $\mathcal{M}$. Quel est son nombre de sommets ? D'arêtes ? Comment expliquez-vous la différence avec les valeurs obtenues sur le graphe de DE BRUIJN qui comprend l'ensemble de tous les mots ?

f) Représentez (en rouge) les chemins correspondant aux mots de $\mathcal{M}$ dans le graphe.

g) Construisez une plus courte superséquence commune aux mots de $\mathcal{M}$.

Exercice 2 : Abordons les bords sans déborder

Le chevauchement de deux mots x et y est la taille du plus long suffixe propre de x qui est également préfixe de y.

Le bord d'un mot x est la taille du chevauchement entre x et x.

Étant donné un mot x, la notation $x[i]$ désigne le i^{e} symbole du mot x ; la notation $x[i..j]$ désigne le facteur commençant au i^{e} symbole et terminant au j^{e} symbole (le facteur $x[i..j]$ est donc de longueur $j - i + 1$).

Algorithme 1 – CalculeBords

```
1  Entrées :
2    mot: Chaîne % chaîne de caractère dont on souhaite calculer les bords de tous
3                  les préfixes %
4    n: Entier   % Longueur de la chaîne mot %
5  Sortie :
6    bord: Tableau de n+1 Entier % Tableau contenant le taille des bords des
7                                  préfixes de la chaîne mot, tel que bord[i]
8                                  correspond à la longueur du bord de mot[1..i]
9                                  si i>0 ou du mot vide si i=0 %
10 Variables :
11   i: Entier % Longueur du préfixe dont on calcule le bord %
12   ℓ: Entier % Longueur du bord de mot[1..i] %
13 Début
14   ℓ ← 0
15   bord[ℓ] ← −1
16   Pour i de 1 à n Faire
17     ℓ ← bord[i − 1] + 1
18     Tant Que ℓ > 0
19       et Que mot[i] ≠ mot[ℓ] Faire
20       ℓ ← bord[ℓ] + 1
21     Fin Tant Que
22     bord[i] ← ℓ
23   Fin Pour
24   ℓ ← 0
25   bord[ℓ] ← 0
26   Retourner bord
27 Fin
```

Question 3 : Écrire la trace de l'algorithme 1 lorsqu'il est appelé avec le mot ROUDOUDOU.

ligne	i	ℓ	$\ell > 0$	$mot[i]$	$mot[\ell]$	$mot[i] \neq mot[\ell]$	$bord[i]$	$bord[\ell]$
15	–	0	Faux	–	–	–	–	−1
16	1			R			–	
17		0	Faux					−1
22							0	
16	2			O			–	
17		1	Vrai		U	Vrai		0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
25	$n+1$	0	Faux	–	–	–	–	0

Question 4 : Complétez le tableau *bord* ci-dessous :

	ε	R	O	U	D	O	U	D	O	U
0	1	2	3	4	5	6	7	8	9	

Question 5 : Expliquez succinctement le principe de l'algorithme 1.

Question 6 : Donnez l'idée générale de la preuve de l'algorithme 1.

Exercice 3 : Des alignements colorés ? Aaaaaaa.

Question 7 : Adaptez l'algorithme d'alignement global par programmation dynamique vu en cours pour déterminer les plus longues sous-séquences communes entre `ATTCAC` et `TATC` (faites apparaître les flèches de *backtracking*).

- coût d'une substitution : $s(x, y) =$ ..... si $x = y$ et ..... si $x \neq y$.
- coût d'un indel : .....
- matrice d'édition :

		A	T	T	C	A	C
T							
A							
C							

- Plus longues sous-séquences communes : .....

Question 8 : En supposant que l'on utilise les mêmes coûts de substitution (match/mismatch), d'insertion et de suppression, les algorithmes d'alignement local et global produisent-ils les mêmes alignements optimaux ? Expliquez.

Question 9 : Complétez la dernière ligne des matrices de match/mismatch, délétion et insertion de l'algorithme avec pénalité de gap affine pour les mots `ATTCAC` et `TAC`, puis déterminez les alignements optimaux (faites apparaître les flèches de *backtracking* en couleur !)

Fonction : $c(g) = -d - (g-1) \times e$ avec $d = 3$, $e = 1$, g : longueur du gap.

Substitution (match/mismatch) :

$$M(i,j) = max \begin{cases} M(i-1, j-1) + s(x_i, y_j) \\ D(i-1, j-1) + s(x_i, y_j) \\ I(i-1, j-1) + s(x_i, y_j) \end{cases}$$

avec $s(a, b) = -1$ si $a \neq b$ et 2 si $a = b$

Matrice M :

	$y_j \rightarrow$	A	T	T	C	A	C
$\downarrow x_i$	0	-3	-4	-5	-6	-7	-8
T	-3	-1	-1	-2	-6	-7	-8
A	-4	-1	-2	-2	-3	-3	-7
C	-5	-5	-2				

Suppression :

$$D(i,j) = max \begin{cases} M(i-1, j) - d \\ D(i-1, j) - e \end{cases}$$

Matrice D :

	$y_j \rightarrow$	A	T	T	C	A	C
$\downarrow x_i$	0	-3	-4	-5	-6	-7	-8
T	-3	-4	-5	-6	-7	-8	-9
A	-4	-4	-4	-5	-8	-9	-10
C	-5	-4	-5				

Insertion :

$$I(i,j) = max \begin{cases} M(i, j-1) - d \\ I(i, j-1) - e \end{cases}$$

Matrice I :

	$y_j \rightarrow$	A	T	T	C	A	C
$\downarrow x_i$	0	-3	-4	-5	-6	-7	-8
T	-3	-4	-4	-4	-5	-6	-7
A	-4	-5	-4	-5	-5	-6	-7
C	-5	-6	-7				

Question 10 : Un mot ayant pour périodes 3 et 5 est-il constitué d'une seule lettre ? Si oui, démontrez-le brièvement. Sinon, trouvez une condition supplémentaire pour que ce soit vrai et démontrez-le brièvement.

Bon Courage. . .