DÉPARTEMENT D'INFORMATIQUE ET DE RECHERCHE OPÉRATIONNELLE

SIGLE DU COURS : IFT 3295/BIN 6000 (Aut2018)

NOM DU PROFESSEUR : Nadia El-Mabrouk

TITRE DU COURS : Bio-Informatique

EXAMEN INTRA

Date : Mardi 30 Octobre 2018
Heure : 13h30 - 15h25
Salle : B-3265 Pav. J.-Brillant

DIRECTIVES PÉDAGOGIQUES :

- **Aucune documentation** n'est permise.
- **Inscrivez tout de suite votre nom et matricule** en bas de cette page.
- Répondez **sur le questionnaire** en utilisant l'espace libre qui suit chaque question.
- Lisez attentivement toutes les questions avant de commencer.
- Les exercices sont indépendants les uns des autres.

1. _____ /18

2. _____ /26

3. _____ /24

4. _____ /32

Total : _____ /100

Nom : _____ **Matricule :** _____

Règlement sur le plagiat : (extrait du règlement disciplinaire sur le plagiat de l'Université de Montréal).
Constitue un plagiat :

1. Faire exécuter son travail par un autre ;
2. Utiliser, sans le mentionner, le travail d'autrui ;
3. Échanger des informations lors d'un examen ;
4. Falsifier des documents.

Le plagiat est passible de sanctions allant jusqu'à l'exclusion du programme.

Exercice 1 : (18 points)

I. Pour chacun des énoncés suivants indiquer s'il est vrai ou non en encerclant OUI ou NON.

1. Soient S et T deux séquences et D la matrice $m \times n$ telle que pour tous $1 \leq i \leq m, 1 \leq j \leq n$, $D(i,j)$ est la distance d'édition (nombre min d'indels et substitutions de caractères) entre $S[1,i]$ et $T[1,j]$. Alors la valeur d'une case de D est toujours supérieure ou égale aux valeurs des trois cases précédentes sur la ligne, la colonne et la diagonale. OUI NON

2. Le score SP d'un alignement $\mathcal{A}$ consistant avec un arbre guide T est égal au score de l'arbre T (i.e. somme des scores des arêtes, où le score d'une arête v est la distance d'édition entre les deux séquences reliées par v). OUI NON

3. Un alignement multiple consistant avec l'arbre étoile a un score SP qui est toujours deux fois supérieur au score SP d'un alignement multiple optimal. OUI NON

4. Soit $\mathcal{A}$ un alignement multiple de score SP minimal pour un ensemble $\{S_i, 1 \leq i \leq m\}$ de séquences. Alors le score induit pour deux séquences S_i, S_j quelconques de cet ensemble est supérieur ou égal à la distance d'édition $D(S_i, S_j)$. OUI NON

5. L'algorithme Knuth-Morris-Pratt pour la recherche exacte de mots est d'autant plus rapide que l'alphabet est grand. OUI NON

6. Une structure secondaire d'ARN peut se représenter par une expression régulière OUI NON

7. Étant donnée une séquence $S[1,n]$, trouver le repliement de S en une seule tige-boucle (une épingle à cheveux) qui maximise le nombre d'appariements Watson-Crick peut se faire par un algorithme de programmation dynamique en temps $O(n^2)$ OUI NON

II. Justifier votre réponse pour un parmi les 7 énoncés précédents.

Justification de l'énoncé numéro :

Exercice 2 : Alignement de séquences (26 points)

1. Dans cette question la valeur de similarité d'un alignement est calculée en attribuant le **score de 2 pour un "match"** (alignement de deux caractères identiques), et le **score de -1 pour une erreur**, c'est-à-dire pour l'alignement de deux caractères différents (mismatch), ou l'alignement d'un nucléotide avec un caractère vide (indel). Nous nous intéressons à **l'alignement global** entre deux séquences u de taille m et v de taille n.

   (a) Expliciter les conditions initiales et écrire la relation de récurrence permettant de calculer la valeur maximale $V(i, j)$ d'un alignement global entre le préfixe $u[1, i]$ de taille i de u et le préfixe $v[1, j]$ de taille j de v, pour tous $1 \leq i \leq m$ et $1 \leq j \leq n$. Expliquer comment retrouver dans la table de programmation dynamique la valeur maximale d'un alignement global entre les deux séquences, ainsi qu'un alignement donnant lieu à cette valeur.

   (b) Remplir la table de programmation dynamique suivante dans l'objectif de trouver un alignement global de score maximal entre $u = CCAC$ et $v = CTTTAC$. **Inclure les flèches** permettant de remonter dans la table.

$V(i,j)$		C	T	T	T	A	C
C							
C							
A							
C							

(c) Quelle est la valeur de similarité entre les séquences u et v de l'exemple ? Expliciter un alignement optimal donnant lieu à cette valeur.

2. Dans cette question, une suite d'insertions consécutives ou une suite de suppressions consécutives sont considérées comme une seule opération (un seul "gap"). Par exemple, l'alignement suivant contient 2 "gaps" et une substitution.

C	T	T	A	-	-	A
C	-	-	A	G	C	C

Nous considérons le score de similarité suivant : 2 pour les "match" et -1 pour les mismatches et "gaps". Par exemple l'alignement ci-dessus a un score de 1.

(a) Expliciter les conditions initiales et écrire les relations de récurrence permettant de calculer $V(i,j)$ pour tout $1 \leq i \leq m$ et tout $1 \leq j \leq n$, **de la façon la plus efficace possible**. Pour celà, on rappelle qu'il est nécessaire de considérer :
- $E(i,j)$: valeur maximale d'un alignement entre $u[1,i]$ et $v[1,j]$ se terminant par l'appariement $(-, v_j)$;
- $F(i,j)$: valeur maximale d'un alignement entre $u[1,i]$ et $v[1,j]$ se terminant par l'appariement $(u_i, -)$;
- $G(i,j)$: valeur maximale d'un alignement entre $u[1,i]$ et $v[1,j]$ se terminant par l'appariement (u_i, v_j) ;

(b) En considérant cette nouvelle valeur de similarité, quelle est la valeur maximale d'un alignement entre les séquences $u = CCAC$ et $v = CTTTAC$? Exhiber TOUS les alignements donnant lieu à cette valeur (on ne demande pas de construire les tables, juste de donner les résultats).

Exercice 3 : Alignement multiple (**24 points**)

On considère un ensemble $\mathcal{S}$ de séquences qu'on veut aligner. On considère la distance d'édition, **qu'on note** D, pour la comparaison des séquences deux à deux.

Supposons qu'un arbre phylogénétique T (binaire, enraciné) soit connu pour $\mathcal{S}$: les feuilles de l'arbre représentent les séquences de $\mathcal{S}$, et les nœuds internes représentent les ancêtres. On rappelle que le problème de **l'alignement phylogénétique soulevé** consiste à étiquetter les nœuds internes de T par des séquences de $\mathcal{S}$, de telle sorte à minimiser le score total de l'arbre (somme des scores de toutes les branches de l'arbre, le score d'une branche étant la distance d'édition entre les deux séquences représentées aux extrémités de la branche). L'arbre étiquetté est appelé un *alignement soulevé* et le score de l'arbre est le *score de l'alignement soulevé*.

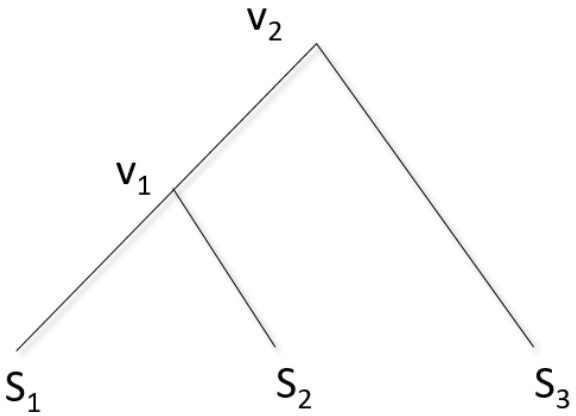
1. Rappeler l'algorithme de programmation dynamique permettant d'inférer un alignement soulevé de score minimal : (1) Relations de récurrence permettant de calculer les valeurs de $d(v, S)$, pour tout nœud v et toute séquence $S \in \mathcal{S}$; $d(v, S)$ représente le score optimal du sous-arbre de racine v sachant que la séquence S est affectée au nœud v ; (2) Comment retrouver le score optimal ; (3) Comment retrouver un alignement soulevé optimal.

2. On considère l'ensemble $E = \{S_1, S_2, S_3\}$ des 3 séquences suivantes :

S_1 : A C C G
S_2 : A C G C
S_3 : A C G G C

Illustrer la méthode de programmation dynamique décrite ci-dessus sur l'arbre ci-dessous. Représenter directement sur l'arbre : (1) Le vecteur à chaque nœud v donnant les valeurs de $d(v, S)$ pour toute séquence $S \in \mathcal{S}$; (2) Le score d'un alignement soulevé optimal ; (3) Une assignation optimale des nœuds internes. Pour commencer, remplire la table suivante contenant la distance d'édition entre chaque paire de séquences (**on ne demande pas d'expliquer comment cette distance d'édition est obtenue**).

$D(i,j)$	S_1	S_2	S_3
S_1			
S_2			
S_3			



3. Expliciter un alignement multiple des séquences S_1, S_2 et S_3 qui soit induit par un alignement consistant avec l'alignement soulevé précédent. Quel est le score SP de l'alignement multiple obtenu ?

Exercice 4 : Recherche exacte (32 points)

Soit Σ un alphabet de taille $|\Sigma|$, T un texte de taille n et P un mot de taille m. Le problème est de rechercher toutes les occurrences de P dans T.

Dans cet exercice, nous utiliserons le texte T et le mot P suivants pour illustrer les méthodes :

$$T = AACAACAAGAACACG\,;\; P = AACAAG$$

Par exemple, le tableau suivant illustre le déroulement de l'algorithme naïf. Les comparaisons effectuées à chaque étape i sont représentées à la ligne i en haut du texte.

4					|	|	|	|	|	*						
3				X												
2			|	x												
1		|	|	|	|	|	x										
	T:	A	A	C	A	A	C	A	A	G	A	A	C	A	C	G
1	P:	A	A	C	A	A	G									
2			A	A	C	A	A	G								
3				A	A	C	A	A	G							
4					A	A	C	A	A	G						

1. Quelle est la complexité en temps, dans le pire des cas, de l'algorithme naïf ?

2. On rappelle que l'algorithme de Knuth-Morris-Pratt effectue les décalages en fonction des plus long bords disjoints du mot P. Autrement dit si à une étape donnée le plus long préfixe de P qui coïcide avec la partie alignée dans le texte est $P[1, i]$, alors le décalage se fera en fonction du plus long suffixe de $P[1, i]$ qui est aussi préfixe de $P[1, i]$, mais suivits de caractères différents.

   (a) Expliciter le décalage effectué pour chaque préfixe du mot P.

P :	A	A	C	A	A	G

(b) Décrire le déroulement de l'algorithme de Knuth-Morris-Pratt dans le tableau ci-dessous. Représenter les comparaisons effectuées à chaque étape ("mismatchs" et occurrences, comme dans l'illustration de la recherche naïve).

4																
3																
2																
1																
	T:	A	A	C	A	A	C	A	A	G	A	A	C	A	C	G
1	P:	A	A	C	A	A	G									
2																
3																
4																

(c) Quelle est la complexité en temps, dans le pire des cas, de l'algorithme de Knuth-Morris-Pratt ?

3. On rappelle que l'algorithme de Horspool effectue ses décalages à partir du caractère du texte aligné avec le dernier caractère du mot. Autrement dit si j est la position dans T telle que $T[j]$ est aligné avec $P[m]$, à l'étape suivante l'algorithme de Horspool considère le plus petit décalage $d(T[j])$ permettant (si possible) de faire coïcider le caractère $T[j]$ du texte avec un caractère de P.

(a) Dans le cas de l'exemple de l'exercice ($P = AACAAG$), expliciter le décalage pour chaque caractère de Σ.

Σ :	A	C	G	T

(b) Écrire, en pseudo-code, l'algorithme de Horspool. On ne demande pas de détailler la phase de prétraitement.

(c) Quelle est la complexité dans le pire des cas de l'algorithme de Horspool ?

(d) Décrire le déroulement de l'algorithme de Horspool dans le tableau ci-dessous.

4															
3															
2															
1															
	T:	A	A	C	A	A	C	A	A	G	A	A	C	A	C	G
1	P:	A	A	C	A	A	G									
2																
3																
4																

4. Supposons que l'on recherche toutes les occurences du mot P dans T à une erreur près. Par exemple, dans ce cas il y a occurrence du mot $P = AACAAG$ à la position 1 du texte T de l'exemple. Expliquer comment il faudrait modifier l'algorithme de Horspool dans ce cas. Plus précisément :

   (a) Expliquer comment modifier la phase de prétraitement, et illustrer votre méthode sur le mot $P = AACAAG$.

(b) Expliquer comment modifier la phase de recherche de l'algorithme de Horspool

(c) Décrire le déroulement de l'algorithme de Horspool ainsi modifié.

4																
3																
2																
1																
	T:	A	A	C	A	A	C	A	A	G	A	A	C	A	C	G
1	P:	A	A	C	A	A	G									
2																
3																
4																