TP 2I003 Recherche des structures secondaires d'une chaîne d'ADN

Une séquence d'ADN (acide Desoxyribonucleique) est une suite de nucléotides, chacun comprenant trois éléments
— une molécule de base : l'adénine A, la guanine G, la cytosine C ou la thymine T,
— un sucre, le désoxyribose,
— et du phosphate.

Chaque nucléotide est désigné par la première lettre de sa molécule de base. Une séquence d'ADN est ainsi un mot sur l'alphabet $\mathcal{A} = \{A, C, G, T\}$. Par exemple, $a = ATCGGCTGCATTTCGA$ représente une séquence d'ADN.

Bien que l'on considère la structure d'une séquence d'ADN comme une suite de nucléotides, sa représentation physique est plus complexe : en effet, les nucléotides se regroupent par paires, l'adényne A avec la thymine T et la guanine G avec la cythosine C.

La structure secondaire S d'une séquence d'ADN $a = a_1 \cdots a_n$ de taille n est définie comme un ensemble de couples d'indices (i, j), avec $1 \leq i < j \leq n$ vérifiant les propriétés suivantes :

1. S définit un couplage : pour tout indice $i \in \{1, \ldots, n\}$, il existe au plus un couple de S contenant i.
2. On suppose que il n'y a pas de nœud : il n'existe pas de couple (i_1, j_1) et (i_2, j_2) de S avec $i_1 < i_2 < j_1 < j_2$.
3. Tout couple $(i, j) \in S$ vérifie $(a_i, a_j) \in \{(A, T), (T, A), (G, C), (C, G)\}$.

Pour tout couple $(i, j) \in S$, on dira que i est couplé avec j et j est couplé avec i.

Le problème que l'on se pose revient à déterminer une structure secondaire S de a dont le nombre de couples est maximum. Par exemple, pour la séquence $a = TCGGCTGCATTTCGA$, une structure secondaire de taille maximale est donnée par $S = \{(1, 15), (2, 14), (3, 13), (4, 8), (5, 7), (9, 10)\}$.

Exercice 1 – Définition et évaluation des algorithmes à programmer

Soit une séquence d'ADN $a = a_1 \ldots a_n$ de taille n. Pour tout couple (i, j) d'indices tels que $1 \leq i \leq j \leq n$ on note $a_{i,j}$ la sous-séquence $a_i \ldots a_j$ de a, $S_{i,j}$ une structure secondaire de a_{ij} dont le nombre de couples est maximum et $E_{i,j} = |S_{i,j}|$ le nombre d'éléments $S_{i,j}$.

Pour l'exemple précédent, $a_{3,13} = GGCTGCATTTC$, $S_{3,13} = \{(3, 13), (4, 8), (5, 7), (9, 10)\}$ et $E_{3,13} = 4$.

Question 1

Un nucléotide ne peut former une paire avec lui même. Que valent $S_{i,i}$ et $E_{i,i}$ pour tout $i \in \{1, \cdots, n\}$?

Question 2

Dans cette question, on considère un couple $(i, j) \in \{1, \cdots, n\}^2$ fixé tel que $i \leq j$. Pour toutes valeurs entières i' et j' telles que $i \leq i' \leq j' \leq j$, on note $T_{i',j'}$ le sous-ensemble de $S_{i,j}$ dont les couples sont à valeurs dans l'intervalle $\{i', \cdots, j'\}$.

1. Supposons tout d'abord que ni i, ni j ne sont couplés dans $S_{i,j}$. Que vaut $S_{i,j}$ en fonction de $T_{i+1,j-1}$? Montrez que $|T_{i+1,j-1}| = |S_{i+1,j-1}|$. En déduire la valeur de $E_{i,j}$ en fonction de $E_{i+1,j-1}$.
2. Supposons que j ne soit pas couplé dans $S_{i,j}$. Que vaut $S_{i,j}$ en fonction de $T_{i,j-1}$? Montrez que $|T_{i,j-1}| = |S_{i,j-1}|$. En déduire la valeur de $E_{i,j}$ en fonction de $E_{i,j-1}$.
3. On suppose maintenant que le couple $(i, j) \in S_{i,j}$. Que vaut $S_{i,j}$ en fonction de $T_{i+1,j-1}$? En déduire que $|S_{i,j}| = 1 + |T_{i+1,j-1}| = 1 + |S_{i+1,j-1}|$. et la valeur de $E_{i,j}$ en fonction de $E_{i+1,j-1}$.
4. On suppose enfin que le couple $(k, j) \in S_{i,j}$ avec $k \in \{i + 1, \cdots, j - 1\}$. Exprimez $S_{i,j}$ en fonction de $T_{i,k-1}$ et $T_{k,j}$. En déduire que $|S_{i,j}| = |T_{i,k-1}| + |T_{k,j}| = |S_{i,k-1}| + |S_{k,j}|$ et la valeur de $E_{i,j}$ en fonction de $E_{i,k-1}$ et $E_{k,j}$.

Question 3
Soit la fonction $e : \{1, \cdots, n\} \times \{1, \cdots, n\} \to \{0, 1\}$ qui vaut 1 ssi i et j peuvent être couplés.

1. Pour simplifier les cas particuliers, on pose $E_{i,j} = 0$ pour $j < i$. Déduire de la question précédente que, pour tout couple $(i, j) \in \{1, \cdots, n\} \times \{1, \cdots, n\}$ avec $i < j$ on a :

$$E_{i,j} = \max(E_{i+1,j-1} + e(i, j), E_{i,j-1}, \max_{i<k<j} E_{i,k-1} + E_{k,j})$$

   Pour cela, vérifier que cette formule est vraie pour tout couple $(i, i + 1)$ pour $i \in \{1, \cdots n - 1\}$, et puis dans le cas ou $i + 1 < j$.

2. En déduire que

$$E_{i,j} = \max(E_{i+1,j-1} + e(i, j), \max_{i<k\leq j} E_{i,k-1} + E_{k,j})$$

   pour tout couple $(i, j) \in \{1, \cdots, n\} \times \{1, \cdots, n\}$ avec $i < j$.

Question 4
En déduire une fonction récursive `tailleMaxRec` qui pour une séquence a de taille n et deux entiers $(i, j) \in \{1, \cdots, n\}$ retourne $E_{i,j}$.

Question 5
Démontrer la validité et la terminaison de la fonction `tailleMaxRec`.

Question 6
Le but de cette question est d'étudier la complexité de la fonction `tailleMaxRec`. Soit u_p le nombre d'appels de la fonction `tailleMaxRec` effectués pour $p = j - i$.

1. Que vaut u_0 ? u_1 ?
2. Démontrez que, pour $p \geq 2$, $u_p = u_{p-2} + 1 + 2\sum_{i=0}^{p-1} u_i$.
3. En déduire que $u_p \geq 2^n$.
4. En déduire l'ordre de grandeur de la complexité de `tailleMaxRec`.

Question 7
Par la suite, on propose un algorithme pour calculer les valeurs $E_{i,j}$ et les stocker dans une matrice E de taille $n \times n$. $E_{i,j}$ désigne dans cet algorithme la valeur stockée en $i$ième ligne et $j$ième colonne.
Le principe de l'algorithme est d'effectuer un calcul en diagonale. Le code de l'algorithme est le suivant :

```
E[1,1] := 0;
For i := 2 To n Do
    E[i,i] := 0
    E[i,i-1] := 0
For p := 1 To n-1 Do
    For i := 1 to n-p Do
        E[i,i+p] = max(E[i+1,i+p-1] + e(i,i+p), max_{i<k≤i+p} E[i,k-1] + E[k,i+p])
```

Quelle est la complexité de cet algorithme ? Justifiez votre réponse. On rappelle que $\sum_{i=1}^{n} i^2 = \frac{n(n+1)(2n+1)}{6}$.

Exercice 2 – Mesure expérimentale de la complexité

Le but de ce second exercice est de mesurer de manière expérimentale la complexité de la fonction `tailleMaxRec` et de l'algorithme décrit dans la question 7 de l'exercice précédent. Le choix du langage de programmation est libre. La complexité expérimentale de l'exécution d'une fonction correspond ici au temps qu'il faut pour que la fonction s'exécute.

Dans cet exercice, n désigne la taille de la séquence initiale.

Question 1
Programmez et testez la fonction `tailleMaxRec`. Programmez également l'algorithme décrit dans la question 7. Appelez cette fonction `tailleMaxIter`. Fournir un jeu d'essai.

Question 2
Programmez et testez la fonction `SeqAleatoire(n)` qui renvoie une séquence d'ADN aléatoire de taille n.

Question 3
On souhaite dans cette question étudier les limites des deux fonctions `tailleMaxRec` et `tailleMaxIter`. Pour cela, exécutez les pour des séquences de taille n de plus en plus importante. Quelle est la la plus grande valeur de n que vous pouvez traiter (sans problème de mémoire, et/ou en un temps raisonnable de quelques minutes) ?

Question 4
On souhaite dans cette question vérifier de manière expérimentale que la fonction `tailleMaxRec` a une complexité exponentielle.

1. Testez la fonction pour des tailles de séquence n de plus en plus grandes. Soit $CRec(n)$ le temps que vous obtenez pour une séquence de taille n. Donnez les valeurs $CRec(n)$ les valeurs de n testées.
2. Vérifiez expérimentalement que $\log CRec(n)$ est une fonction linéaire de n (pour les valeurs importantes de n) en calculant la pente de la droite.

Question 5
On souhaite maintenant vérifier de manière expérimentale que la fonction `tailleMaxIter` a une complexité polynomiale $\Theta(n^\alpha)$ avec $\alpha \in \mathbb{N}^\star$ déterminé dans la question 7 de l'exercice 1.

1. Testez la fonction pour des valeurs n de plus en plus grandes. Soit $CIter(n)$ le temps que vous obtenez pour une séquence de taille n. Donnez les valeurs $CIter(n)$ pour les valeurs de n testées.
2. Vérifiez expérimentalement que $\frac{Citer(n)}{n^\alpha}$ est une fonction constante quand n croît (pour les valeurs importantes de n) en calculant cette valeur.