

TD n°7

TD noté

Ce TD note est a realiser individuellement. Toute communication est interdite. L'epreuve dure deux heures. La seule documentation autorisee est une feuille A4 manuscrite. Toute autre documentation ou appareil electronique est interdit. L'utilisation de telephones portables est interdite, même comme montre.

Premier problème : arbres k -aires

On considere un arbre k -aire : chaque nœud N a k fils $N:\text{fils}(1), N:\text{fils}(2), \dots, N:\text{fils}(k)$. N stocke aussi une valeur $N:\text{val}$. $k \geq 2$ est fixe. De plus l'arbre est *complet* : il a hauteur h et toutes les feuilles sont a distance h de la racine.

On a le programme suivant :

```
procédure ParcourInfixe(N : noeud) {  
  Si N n'est pas une feuille  
    ParcourInfixe( N.fils(1) )  
  Traite( N.val )  
  Si N n'est pas une feuille  
    ParcourInfixe( N.fils(k) )  
}
```

On note $T(n)$ le temps d'execution de `ParcourInfixe(Racine)` en fonction du nombre n de nœuds et $f(n)$ celui de `Traite`.

Exercice 1 Donner une relation de type "master theorem" exprimant $T(n)$ (repondre en une ligne)

Exercice 2 La resoudre dans les trois cas suivants, selon que l'on suppose que $f(n)$ vaut

{ $f(n) = O(1)$
{ $f(n) = O(\log n)$
{ $f(n) = O(n)$

Attention, le resultat peut dependre de k .

Exercice 3 Tombe-t-on parfois dans le cas "moyen" du master theorem ? Si oui, donner une valeur de $f(n)$ et dire ce que vaut alors $T(n)$. Si non, pourquoi ?

Deuxième problème : distance d'édition en génomique

On considere qu'un **gène** est un mot sur l'alphabet $\{A; T; G; C\}$.

On considere trois operations de base permettant de transformer un gene en un autre :

1. *Insertion* d'une lettre. Exemple AAA devient AATA par insertion d'un T en troisieme position
2. *Suppression* d'une lettre. ATCC devient ACC par suppression d'un T en deuxieme position
3. *Substitution*. Exemple AAAT devient ACAT en substituant un C au deuxieme A. .

Une **séquence d'édition** entre un gene A et un gene B est une suite $A = G_0; G_1; G_2:::G_{k-1}; G_k = B$ de genes telle que l'on passe de G_i à G_{i+1} par l'une des trois operations elementaires autorisees (insertion, suppression, substitution). k est la longueur de la sequence d'edition.

La **distance d'édition** (aussi appelee distance de Levenshtein) entre deux genes A et B est la longueur d'une plus petite sequence d'edition entre A et B .

On se propose de calculer la distance d'edition entre deux genes A et B par recurrence sur leur longueur. Posons quelques notations :

{ $A[i]$ est la i eme lettre du gene A

{ $A[1::i]$ est le gene constitue des i premieres lettres de A

{ $n = |A|$ est le nombre de lettres de A et $m = |B|$ celui de B .

{ $dp(i;j)$ est la distance d'edition entre $A[1::i]$ et $B[1::j]$.

Ainsi ce que l'on veut est calculer $dp(n;m)$.

Par exemple pour $A=ATATCCCG$ et $B = ATTTTCGC$ on a la suite de genes

$ATATCCCG \rightarrow ATTTCCCG \rightarrow ATTTTCGC \rightarrow ATTTTCGC$

(substitution de A par T ; substitution de C par G ; suppression G) donc $dp(8;7) = 3$.

Exercice 4 Que vaut $dp(0;0)$?

Exercice 5 Montrer que si $A[i] = B[j]$ alors $dp(i;j) = dp(i-1;j-1)$

Exercice 6 Montrer que si $A[i] \neq B[j]$ alors $dp(i;j) = 1 + \min(dp(i-1;j); dp(i;j-1); dp(i-1;j-1))$.

Exercice 7 En deduire une fonction recursive $dp(\text{entier } i, \text{entier } j, \text{gene } A, \text{gene } B)$ calculant $dp(i;j)$.

Exercice 8 Donner un pire cas pour le nombre d'appels recursifs effectues par $dp(n,m,A,B)$.

Exercice 9 En remarquant qu'un grand nombre d'appels recursifs sont identiques, et en utilisant un tableau de memorisation, donner une version de dp qui marche en temps polynomial.

Exercice 10 Preciser les complexites en temps et en espace de ce nouvel algorithme.

Exercice 11 On veut aussi retrouver la sequence d'edition correspondant a la longueur calculee. Modifier votre algorithme pour qu'il affiche la sequence qui transforme A en B . L'affichage ressemblera, par exemple, a : \Insertion de T en 2eme position. Substitution de C par G en 4eme position. Suppression de T en 5eme position".

Exercice 12 On prend maintenant une modelisation plus realiste au niveau biologique. Les operations ont un *coût* :

{ Substituer un 'A' par un 'T', ou un 'T' par un 'A', ou un 'C' par un 'G', ou un 'G' par un 'C', coûte 1.

{ Les autres substitutions (entre l'ensemble {'A','T'} et l'ensemble {'C','G'}) donc coûtent 3.

{ Insertions et suppressions coûtent 5.

Le coût d'une sequence d'edition est la somme des coûts de ses operations elementaires. La distance d'edition entre deux genes est maintenant le coût minimum d'une sequence d'edition.

Expliquer rapidement les modifications a apporter a l'algorithme de pour qu'il resolve cette variante du probleme (on veut juste le coût, pas la sequence d'edition).