

Algorithmes et Structures de Données

Vendredi 17 novembre 2023

Durée 1H30 – Cours et TD NON autorisés

1. Fonction d'Ackermann - 6 pts

```

Fonction Ack (m, n : entier) : entier
Var r : entier
Début
  Si m=0
    Alors r ← n+1
    Sinon Si n=0
      Alors r ← Ack(m-1, 1) {@1}
      Sinon r ← Ack(m-1, Ack(m, n-1) { @2}) { @3}
    FinSi
  FinSi
Retourner(r)
Fin

```

Simuler les **états successifs de la pile** pour l'appel de l'instruction écrire(Ack(2, 1)) { @0 }

2. Calcul d'une fraction continue - 6 pts

On suppose donnés $n+1$ nombres entiers strictement positifs a_0, a_1, \dots, a_n dans un tableau a ($a[i] = a_i, i = 0 \dots n$). Type $a = \text{tableau}[0..100]$ d'entier

On souhaite calculer, pour un tableau a et une taille n donnés, la valeur du nombre :

$$x = a_0 + \frac{1}{a_1 + \frac{1}{a_2 + \frac{1}{\dots + \frac{1}{a_n}}}}$$

2.1 - Ecrire une **fonction itérative** qui calcule la valeur de ce nombre.

2.2 - Ecrire une **fonction récursive** qui calcule la valeur de ce nombre. Expliquer le principe récursif.

3. Recherche de facteurs - 8 pts

On dispose de n chaînes de caractères qu'on souhaite mettre bout à bout (avec un chevauchement de longueur fixe k) dans le bon ordre pour reconstituer une chaîne globale (version très simplifiée d'un des problèmes de séquençement de l'ADN).

On suppose qu'il n'y a pas 2 chaînes qui ont le même préfixe ou le même suffixe.

Exemple de chevauchement avec $k=3$:

A G G C T A **A C G** et **A C G** T G A C A T → A G G C T A **A C G** T G A C A T

Exemple : $n=4$, $k=3$

Soient 4 chaînes :

C A T T G A C T A	num=3 (pour le point 3.4)
C T A C C T A T G	num=4
A C G T G A C A T	num=2
A G G C T A A C G	num=1

Reconstitution :

A G G C T A **A C G** T G A **C A T** T G A **C T A** C C T A T G

(les chevauchements sont mis en gras pour mieux comprendre)

3.1. Ecrire en pseudo-langage une procédure qui charge le fichier de nom `nchaines.txt` contenant les n chaînes (une par ligne) dans un tableau `t` de type `tab` (les champs `val` contiennent les chaînes et les champs `num` sont mis à 0)

```
Type case = Enregistrement
           val : chaîne
           num : entier
FinEnregistrement
tab = tableau[1..max] de case {max ≥ n}
```

3.2. Ecrire en pseudo-langage les fonctions `préfixe(c, k)` et `suffixe(c, k)`, qui renvoie la chaîne de caractères correspondant au préfixe (ou suffixe) de longueur k de la chaîne `c`.

3.3. Ecrire en pseudo-langage une procédure qui, étant donné le tableau `t` de n chaînes et le chevauchement k , renvoie une chaîne reconstituée possible `c` et remplit les champs `num` de `t` avec l'ordre d'apparition de la chaîne dans `c` (respectivement 3, 4, 2 et 1 pour l'exemple). Expliquer la méthode en français.

3.4. Ecrire en pseudo-langage une procédure qui enregistre les résultats dans un nouveau fichier (nommé `resultat.txt` et formaté comme ci-dessous) contenant sur la première ligne la chaîne `c` reconstituée et ensuite les n chaînes (une par ligne) avec en fin de ligne la valeur de `num` correspondante.

```
A G G C T A A C G T G A C A T T G A C T A C C T A T G
C A T T G A C T A ; 3
C T A C C T A T G ; 4
A C G T G A C A T ; 2
A G G C T A A C G ; 1
```