GIS3 — Structures de Données — Épreuve d'Avril 2015
Durée 2 heures — Aucun document autorisé

---

1 Petits exercices (8 points)

Question 1 [1 pt]. Soient A une matrice $n \times n$, d'éléments notés a_{ij}, et α un réel. On s'intéresse à la complexité $f(n)$ de l'algorithme suivant. On compte le nombre de multiplications par α. Donner un équivalent asymptotique simple (sous la forme d'un monôme) de $f(n)$.

```
for j variant de 1 à n do
  for i variant de 1 à j do
    a_ij := α a_ij
  end do
end do
```

Question 2 [1 pt]. Donner des instructions C qui permettent de doubler la taille de la zone allouée à s. La chaîne de caractères présente dans s doit être préservée. Il ne doit pas y avoir de fuite mémoire. Vous pouvez utiliser des variables auxiliaires.

```
{   char* s;    /* pointe vers une zone allouée dynamiquement */
    int alloc;  /* le nombre de caractères alloués à s */
    int length; /* la longueur de la chaîne présente dans s (length + 1 < alloc) */
```

Question 3 [1 pt]. Donner la nature de l'erreur relevée par `valgrind` en Figure 1.

Question 4 [2 pts]. Écrire une fonction `ajouter_en_queue`, paramétrée par une liste L (en mode donnée/résultat), un double d, et qui ajoute d en queue de L. Les déclarations à utiliser sont les suivantes.

```
struct maillon {
    double valeur;
    struct maillon* suivant;
};
#define NIL (struct maillon*)0
```

```
struct liste {
    struct maillon* tete;
    int nbelem;
};
```

Question 5 [1 pt]. Insérer les entiers suivants dans un ABR initialement vide. Indiquer les différents états de l'ABR.

37, 18, 109, 88, 100, 40.

```
==5115== Memcheck, a memory error detector
==5115== Copyright (C) 2002-2013, and GNU GPL'd, by Julian Seward et al.
==5115== Using Valgrind-3.10.0.SVN and LibVEX; rerun with -h for copyright info
==5115== Command: ./a.out
==5115==
liste L = [B + C + 2 D -> A + C, 2 B -> B + C, A -> B + 2 D]
complexe numero 2: B + 2 D
complexe numero 4: B + C
complexe numero 5: B + C + 2 D
complexe numero 3: 2 B
complexe numero 1: A
complexe numero 6: A + C
==5115==
==5115== HEAP SUMMARY:
==5115==     in use at exit: 440 bytes in 5 blocks
==5115==   total heap usage: 13 allocs, 8 frees, 952 bytes allocated
==5115==
==5115== LEAK SUMMARY:
==5115==    definitely lost: 88 bytes in 1 blocks
==5115==    indirectly lost: 352 bytes in 4 blocks
==5115==      possibly lost: 0 bytes in 0 blocks
==5115==    still reachable: 0 bytes in 0 blocks
==5115==         suppressed: 0 bytes in 0 blocks
==5115== Rerun with --leak-check=full to see details of leaked memory
==5115==
==5115== For counts of detected and suppressed errors, rerun with: -v
==5115== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
```

FIGURE 1 – Exécution d'un processus sous `valgrind`.

Question 6 [2 pts]. Écrire une fonction, paramétrée par un arbre binaire, qui retourne la somme des éléments de l'arbre. Si l'arbre est vide, la somme doit être nulle. Les déclarations à utiliser sont les suivantes.

```
struct noeud {
    int valeur;
    struct noeud* gauche;
    struct noeud* droit;
};
```

2 Problème (12 points)

On cherche à concevoir des structures de données dédiées à l'analyse de systèmes de réactions chimiques. On suppose l'existence du fichier donné Figure 2.

```
/*
 * N = le nombre maximal d'espèces chimiques autorisé
 *
 * Un complexe est défini par le tableau de ses coefficients stoechiométriques
 * coeff[0] = le coefficient stoechiométrique de A
 * coeff[1] = celui de B
 *         ...
 * coeff[N-1] = celui de la dernière espèce
 */

#define N 10

struct complexe {
    int coeff [N];
};
```

FIGURE 2 – Le fichier `complexe.h`.

Question 7 [3 pts]. On veut implanter les réactions ainsi que les listes chaînées de réactions. Pour chacun de ces types, définir un fichier d'entête, la déclaration C ainsi que les directives d'inclusion à placer dans les fichiers d'entête.

On cherche à construire la matrice d'adjacence M du graphe FHJ d'un système $\mathscr{S}$ de réactions chimiques. Cette matrice est indicée en lignes et en colonnes, par les complexes du système $\mathscr{S}$. Notons n le nombre de complexes de $\mathscr{S}$. Il s'agit donc d'attribuer un numéro appartenant à l'intervalle $[0, n-1]$, à chaque complexe de $\mathscr{S}$. Ce numéro servira d'indice dans M.

On cherche donc une structure de données, une sorte de dictionnaire de complexes, qui permette d'attribuer un numéro à tous les complexes qu'elle contient. Une fois cette structure remplie, on veut pouvoir retrouver efficacement le numéro à partir du complexe.

Deux approches possibles sont listées ci-dessous.

1. Le programme qui utilise la structure de données procède en deux temps : il commence par enregistrer les complexes dans la structure sans se soucier de leur numéro. Quand tous les ajouts sont finis, il appelle une fonction/action chargée de numéroter tous les complexes enregistrés.
2. Un numéro est attribué à chaque complexe, dès son ajout dans la structure. La gestion du numéro est interne à la structure de données (la fonction ou l'action qui utilise la structure n'a pas besoin de gérer une variable locale `numero`, distincte de la structure).

Question 8 [1 pt]. Décrire votre solution dans ses grandes lignes : quelle approche choisissez-vous ? quelle implantation de dictionnaire choisissez-vous, parmi celles qui ont été étudiées en cours ?

Question 9 [2 pts]. Préciser votre solution : donner les déclarations de type et leurs spécifications.

Question 10 [2 pts]. Si vous avez choisi la première approche, donner la fonction/action qui numérote les complexes quand les ajouts sont finis. Si vous avez choisi la seconde approche, donner la fonction/action qui effectue l'ajout. Dans les deux cas, vous pouvez définir des fonctions auxiliaires.

Question 11 [2 pts]. Traduire en C, en utilisant votre solution, le pseudo-code de la Figure 3. Vous pouvez supposer que M est de dimension `NMAX`$\times$`NMAX` où `NMAX` est une constante supposée définie.

```
procedure matrice_adjacence_FHJ (M, L)
  M est une matrice initialement remplie de zéros (donnée/résultat)
  L est une liste de réactions (donnée)
begin
  Soit D une variable locale ayant le type que vous avez défini en question 9
  Initialiser D à vide
  for chaque réaction r de L do
    Ajouter à D le réactant et le produit de r (deux complexes)
  end do
  Si nécessaire, numéroter les complexes présents dans D
  for chaque réaction r de L do
    i := le numéro, dans D, du réactant de r
    j := le numéro, dans D, du produit de r
    M[i][j] := 1
  end do
end
```

FIGURE 3 – Pseudo-code construisant la matrice d'adjacence M du graphe FHJ d'un système de réactions chimiques.

Question 12 [2 pts]. Pour tester si le théorème de déficience zéro s'applique à un système de réactions chimiques $\mathscr{S}$, il faut calculer le nombre de composantes connexes du graphe FHJ et tester si ces composantes connexes sont fortement connexes. Ces deux tests peuvent être effectués par des parcours en profondeur d'abord sur le graphe FHJ. Ces parcours peuvent se réaliser en utilisant des piles. Donner la déclaration et les spécifications d'un type pile adapté à ces parcours.

Document préparatoire à l'épreuve écrite

Dans ce document, il est surtout important de se familiariser avec les expressions en italique. Vous pouvez aussi revoir les quelques notions de théorie des graphes mentionnées.

Les systèmes de réactions chimiques généralisées fournissent un formalisme simple de modélisation des systèmes dynamiques. Un livre de référence est [1]. Le système suivant est une variante d'un modèle inventé par Alfred Lotka dans les années 1920 et Vito Volterra.

$$H + L \longrightarrow H + 2\,L\,, \tag{1}$$
$$L + Y \longrightarrow 2\,Y\,, \tag{2}$$
$$Y \longrightarrow \varnothing\,. \tag{3}$$

Les *espèces* chimiques sont désignées par les lettres H, L et Y. Elles représentent des quantités d'herbe, de lapins et de lynx[1]. Le système est composé de trois *réactions* : (1), (2) et (3). Les expressions $H + L$, $H + 2\,L$, $L + Y$, $2\,Y$, Y et $\varnothing$ sont appelées des *complexes*. Elles ont l'apparence de combinaisons linéaires d'espèces chimiques. Les coefficients sont des entiers. Ce sont les *coefficients stœchiométriques*. Par exemple, le coefficient stœchiométrique de l'espèce L dans le complexe $H + 2\,L$ est égal à 2 ; le coefficient stœchiométrique de l'espèce H est égal à 1. Le complexe qui apparaît en partie gauche d'une réaction est le *réactant* de la réaction. Le complexe qui apparaît en partie droite est le *produit* de la réaction.

En termes un peu techniques, la réaction (1) décrit la dynamique suivante : lorsqu'une « molécule » d'herbe rencontre une « molécule » de lapin, il est possible qu'une réaction se déclenche, consomme ces deux molécules, et produise une « molécule » d'herbe et deux « molécules » de lapin. Le nombre de « molécules » d'herbe est donc conservé. Le nombre de « molécules » de lapins augmente de 1. Bien sûr, le terme « molécule » doit être compris dans un sens abstrait et pourrait être traduit par « une certaine quantité ». En Français courant, la réaction (1) modélise l'idée qu'en présence d'herbe, les lapins ont tendance à se reproduire et que l'herbe est en quantité suffisante pour ne pas être affectée par les lapins. La réaction (2) modélise l'idée que lorsqu'on met des lapins et des lynx en présence, les lapins ont tendance à disparaître et les lynx, à se reproduire. Enfin, la réaction (3) exprime l'idée que pour une raison non modélisée (maladie, chasse, grand âge), les lynx peuvent disparaître.

La modélisation par systèmes de réactions chimiques est utilisée en écologie (comme ci-dessus), en épidémiologie (les espèces chimiques sont alors des populations de gens malades, susceptibles de tomber malades, et immunisés), en marketing, etc.

Si on enrichit un système de réactions chimiques $\mathscr{S}$ avec certaines données numériques (population initiale pour chaque espèce, vitesse pour chaque réaction), il est possible de simuler numériquement l'évolution des populations d'espèces chimiques dans le temps. On peut alors déterminer l'état futur du système de réactions chimiques. Bien sûr, cet état futur dépend des informations quantitatives qu'on a dû préciser.

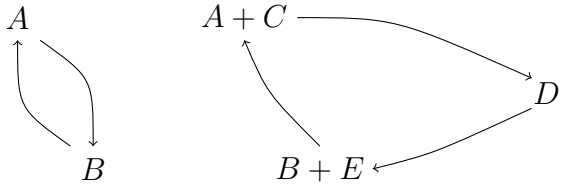
1. On utilise le vocabulaire de la chimie, bien qu'on n'ait pas affaire du tout à des substances réagissant dans des éprouvettes.

Il est aussi possible d'étudier le système $\mathscr{S}$ du point de vue de la théorie des graphes et d'obtenir ainsi des informations partielles sur l'état futur du système $\mathscr{S}$ qui sont valables quelles que soient les données numériques. Un exemple est donné par le « théorème de déficience zéro » qu'il serait trop compliqué d'expliquer précisément ici.

Un graphe souvent associé à un système de réactions chimiques $\mathscr{S}$ est le *graphe FHJ* (du nom de Feinberg, Horn et Jackson) du système. Il s'agit d'un graphe orienté dont les sommets sont les complexes de $\mathscr{S}$. Les arcs sont donnés par les réactions. Par exemple, le système suivant :

$$A \longrightarrow B\,, \quad A+C \longrightarrow D\,, \quad B+E \longrightarrow A+C\,, \quad B \longrightarrow A\,, \quad D \longrightarrow B+E\,,$$

a pour graphe FHJ :



La formule de la déficience d'un système $\mathscr{S}$ dépend du nombre de composantes connexes du graphe FHJ. Avant d'appliquer le théorème de déficience zéro, il est utile de savoir si toute composante connexe est fortement connexe. Ces analyses peuvent être effectuées à partir de n'importe quelle représentation du graphe, comme la matrice d'adjacence du graphe.

Voici une des représentations les plus simples possibles des complexes. Un code C permettant de lire et d'imprimer un complexe est disponible sur le site du cours.

```
/*
 * N = le nombre maximal d'espèces chimiques autorisé
 *
 * Un complexe est défini par le tableau de ses coefficients stoechiométriques
 * coeff[0] = le coefficient stoechiométrique de A
 * coeff[1] = celui de B
 *         ...
 * coeff[N-1] = celui de la dernière espèce
 */

#define N 10

struct complexe {
    int coeff [N];
};
```

Références

[1] Péter Érdi and János Tóth. *Mathematical models of chemical reactions : theory and applications of deterministic and stochastic models.* Princeton University Press, 1989.