

Éléments de correction
Examen de 2^{nde} session - 24 juin 2010

Exercice 1 : Cours (4 points)

Pour cet exercice, notation QCM : réponse fausse = des points en moins.

Q 1.1 La recherche du premier élément inséré se fait en 1 comparaison puisqu'il n'y a pas eu de collision. La recherche d'un des $n - 1$ éléments insérés ensuite nécessite une ou deux comparaisons puisque soit la première alvéole testée contient l'élément, soit elle ne le contient pas et la seconde alvéole testée contient l'élément recherché. En conclusion, la recherche est en temps constant quel que soit l'élément recherché, complexité en $\Theta(1)$.

Q 1.2 Dans ce cas, toutes les alvéoles contenant au moins un élément en contiennent deux sauf une. On aura encore une complexité en temps constant.

Exercice 2 : Ensembles (8 points)

Q 2.1 Donner le contenu du tableau si MAX vaut 10 et qu'il existe deux ensembles : $\{5, 6, 2\}$ ayant pour représentant 2, et $\{3, 9, 1\}$ ayant pour représentant 9.

9	0	9	-1	2	2	-1	-1	0	-1
---	---	---	----	---	---	----	----	---	----

Q 2.2

```

1  procedure afficher (t : TABLEAU_D_ENSEMBLES);
2  var
3      i, j : INTEGER;
4  begin
5      for j := low(t) to high(t) do begin
6          if t[j] = 0 then begin
7              write('*'); write(j); write('␣');
8              for i := low(t) to high(t) do begin
9                  if t[i] = j then begin
10                     write(i); write('␣');
11                 end {if};
12             end {for};
13             writeln;
14         end {if};
15     end {for};
16 end {afficher};

```

Résultat :

```

1  *2 5 6
2  *9 1 3

```

Q 2.3 On ne déclare que deux entiers.

Q 2.4 Le pire des cas est lorsqu'on a MAX ensembles de 1 élément. On fait ici exactement MAX^2 comparaisons. Soit une complexité en $\Theta(MAX^2)$. Le meilleur des cas est lorsqu'on n'a aucun ensemble. Seule la première boucle est réalisée, soit $\Theta(MAX)$.

Q 2.5 Cette question est la plus difficile de tout l'examen. Il s'agit ici de construire une structure de données dans un tableau intermédiaire. Cette structure sert à lier les éléments d'un même ensemble. On utilisera un tableau `ref` qui donne en i la position du prochain élément de l'ensemble auquel i appartient. Sur l'exemple on aura :

3	0	9	-1	6	2	-1	-1	0	-1
---	---	---	----	---	---	----	----	---	----

 ce qui permet de savoir que 1 va avec 3, 3 avec 9 et 9 est le représentant, ou bien encore que 5 va avec 6, 6 va avec 2 et 2 est le représentant. L'affichage des ensembles se fait en parcourant ce tableau `ref`.

```

1  procedure afficher_bis (t : TABLEAU_D_ENSEMBLES);
2  var
3      ref : array[1..MAX] of INTEGER;
4      last : array[1..MAX] of INTEGER;
5      i, j, k : INTEGER;
6  begin
7      // initialisation de last
8      for i := low(t) to high(t) do
9          last[i] := 0;
10     // calcul de ref
11     for i := low(t) to high(t) do begin
12         if t[i] <= 0 then begin
13             ref[i] := t[i];
14         end else if t[i] > 0 then begin
15             if last[t[i]] > 0 then
16                 ref [last[t[i]]] := i;
17             last[t[i]] := i;
18         end {if};
19     end {for};
20     // finalisation
21     for i := low(t) to high(t) do
22         if last[i] > 0 then
23             ref[last[i]] := i;
24     // affichage des ensembles
25     for j := low(t) to high(t) do begin
26         i := j;
27         if ref[i] > 0 then begin
28             while i > 0 do begin
29                 if ref[i] >= 0 then begin
30                     if ref[i] = 0 then write('*');
31                     write(i); write(' ');
32                     k := ref[i];
33                     { mise a zero pour eviter de reparcourir l'ensemble }
34                     ref[i] := 0;
35                     i := k;
36                 end {if};
37             end {while};
38             writeln;
39         end {if};
40     end {for};
41 end {afficher_bis};

```

Résultat:

```

1  1 3 *9
2  5 6 *2

```

Q 2.6 On utilise deux tableaux de longueur `MAX`.

Q 2.7 Les trois premières boucles `for` s'exécutent en $\Theta(\text{MAX})$. La dernière est une double boucle mais le nombre d'éléments parcourus par i et j est au maximum `MAX`.

Q 2.8 Comme on n'a pas de contrainte d'espace, on peut faire cela très brutalement. On utilise un tableau carré de booléens `u` de taille `MAX` \times `MAX`, on construit `u` à partir les valeurs de `t` de telle sorte que `u[i, j]` est

vrai si j est un élément de l'ensemble représenté par i . Il suffit ensuite de retranscrire u dans τ .

Q 2.9 Il n'y a aucune comparaison d'éléments.

Q 2.10

- $\tau_2[i]$ doit valoir -1 également.
- Parce que si $\tau_1[i] = 0$ alors i ne peut pas appartenir à un ensemble plus grand.
- Parce que si $\tau_1[i] > 0$ et $\tau_1[i] < \tau_2[i]$, alors i appartient à un ensemble dont le représentant est plus petit dans τ_1 que dans τ_2 et donc il le sous-ensemble est plus grand dans τ_1 que dans τ_2 .
- C'est le cas contraire de ci-dessus, l'ensemble dans τ_1 est plus petit que l'ensemble dans τ_2 , ils sont donc inclus si ils ont même représentants dans τ_2 .

Q 2.11 On met sous forme normale puis on applique les règles de la question précédente.

Q 2.12 Le cas qui maximise le nombre de comparaisons est le cas d. Il survient au maximum une fois sur deux lorsqu'on a un ensemble d'ensembles de 2 éléments que l'on compare à un ensemble avec tous les éléments. Par exemple : $\{\{1, 2\}, \{3, 4\}, \{5, 6\}\}$ et $\{\{1, 2, 3, 4, 5, 6\}\}$.

Q 2.13 $\Theta(2 \times \text{MAX})$

Exercice 3 : Complexité de la multiplication de matrices (4 points)

Q 3.1 8 opérations de multiplications.

Q 3.2

$$T(n) = 8 \times T\left(\frac{n}{2}\right)$$

Q 3.3 $n^{\log_2(8)} = n^3$ et donc $f(n) = 0 = \mathcal{O}(n^{3-\epsilon})$, ainsi $T(n) = \Theta(n^3)$.

Q 3.4 7 opérations de multiplications.

Q 3.5

$$T(n) = 7 \times T\left(\frac{n}{2}\right)$$

$n^{\log_2(7)} = n^{2.81}$ et donc $f(n) = 0 = \mathcal{O}(n^{2.81-\epsilon})$, ainsi $T(n) = \Theta(n^{2.81})$.

Exercice 4 : Palindrome (4 points)

Q 4.1 Si $p(i, j)$ est vrai alors $v_i \dots v_j$ est un palindrome.

Q 4.2

	m	o	n		k	a	y	a	k
m	V	F	F	F	F	F	F	F	F
o	F	V	F	F	F	F	F	F	F
n	F	F	V	F	F	F	F	F	F
	F	F	F	V	F	F	F	F	F
k	F	F	F	F	V	F	F	F	F
a	F	F	F	F	F	V	F	F	F
y	F	F	F	F	F	F	V	F	F
a	F	F	F	F	F	V	F	V	F
k	F	F	F	F	V	F	F	F	V

Q 4.3

```

1 var
2   t : array of array of BOOLEAN;
3 begin
4   setlength(t,m+2);
5   for i := 0 to m+1 do begin
6     setlength(t[i],m+2);
7   end {for};

```

Q 4.4

```

1   for i := m downto 1 do begin
2     for j := i+2 to m do begin
3       if v[i] = v[j] then begin
4         t[i][j] := t[i+1][j-1];
5       end else begin
6         t[i][j] := FALSE;
7       end {if};
8     end {for};
9   end {for};

```

Q 4.5

```

1   for i := m downto 1 do begin
2     for j := 1 to m do begin
3       if (i <> j) and (t[i][j]) and not (t[i-1][j+1]) then begin
4         write(i); write(' '); writeln(j);
5       end {if};
6     end {for};
7   end {for};

```