



L1 Informatique et Electronique - SI1: Algorithmique et complexité expérimentale
Première session 2017-2018

2 heures - Tous documents autorisés

Le sujet comporte 6 pages. Le barème est donné à titre indicatif.

QCM (questions 1 à 8) : noircir les cases correspondant aux bonnes réponses sur le sujet.

Partie rédactionnelle (question 9 et 10) : à rédiger sur la copie anonymisée et dérouler la fonction sur le sujet.

Dans le QCM chaque question a exactement une bonne réponse. Une bonne réponse compte pour 1 point et une mauvaise réponse pour -0,5 point, sauf pour la question portant sur la fonction `tri` pour laquelle une bonne réponse vaut 2 points et une mauvaise réponse -1 point.

Pour chaque question, **noircissez** (comme ceci ☒) et non comme cela ☐) la case correspondant à la bonne réponse.

<input type="checkbox"/> 0	<input type="checkbox"/> 0	<input type="checkbox"/> 0	<input type="checkbox"/> 0	<input type="checkbox"/> 0	<input type="checkbox"/> 0	<input type="checkbox"/> 0	<input type="checkbox"/> 0
<input type="checkbox"/> 1	<input type="checkbox"/> 1	<input type="checkbox"/> 1	<input type="checkbox"/> 1	<input type="checkbox"/> 1	<input type="checkbox"/> 1	<input type="checkbox"/> 1	<input type="checkbox"/> 1
<input type="checkbox"/> 2	<input type="checkbox"/> 2	<input type="checkbox"/> 2	<input type="checkbox"/> 2	<input type="checkbox"/> 2	<input type="checkbox"/> 2	<input type="checkbox"/> 2	<input type="checkbox"/> 2
<input type="checkbox"/> 3	<input type="checkbox"/> 3	<input type="checkbox"/> 3	<input type="checkbox"/> 3	<input type="checkbox"/> 3	<input type="checkbox"/> 3	<input type="checkbox"/> 3	<input type="checkbox"/> 3
<input type="checkbox"/> 4	<input type="checkbox"/> 4	<input type="checkbox"/> 4	<input type="checkbox"/> 4	<input checked="" type="checkbox"/> 4	<input type="checkbox"/> 4	<input type="checkbox"/> 4	<input type="checkbox"/> 4
<input type="checkbox"/> 5	<input type="checkbox"/> 5	<input type="checkbox"/> 5	<input type="checkbox"/> 5	<input type="checkbox"/> 5	<input type="checkbox"/> 5	<input type="checkbox"/> 5	<input type="checkbox"/> 5
<input type="checkbox"/> 6	<input type="checkbox"/> 6	<input type="checkbox"/> 6	<input type="checkbox"/> 6	<input type="checkbox"/> 6	<input type="checkbox"/> 6	<input type="checkbox"/> 6	<input type="checkbox"/> 6
<input type="checkbox"/> 7	<input type="checkbox"/> 7	<input type="checkbox"/> 7	<input type="checkbox"/> 7	<input type="checkbox"/> 7	<input type="checkbox"/> 7	<input type="checkbox"/> 7	<input type="checkbox"/> 7
<input type="checkbox"/> 8	<input type="checkbox"/> 8	<input type="checkbox"/> 8	<input type="checkbox"/> 8	<input type="checkbox"/> 8	<input type="checkbox"/> 8	<input type="checkbox"/> 8	<input type="checkbox"/> 8
<input type="checkbox"/> 9	<input type="checkbox"/> 9	<input type="checkbox"/> 9	<input type="checkbox"/> 9	<input type="checkbox"/> 9	<input type="checkbox"/> 9	<input type="checkbox"/> 9	<input type="checkbox"/> 9

← **Noircir** les cases correspondant à votre numéro d'étudiant ci-contre. Par sécurité, recopiez le ci-dessous ainsi que votre numéro d'anonymat.

Numéro étudiant :

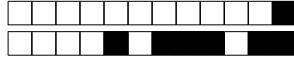
.....

Numéro d'anonymat:

.....

Vous trouverez le numéro d'anonymat sur votre étiquette autocollante





Question 1 Si N est la taille du tableau t . Quelle est la complexité au pire cas, la plus précise, pour la fonction `tri`?

```
void tri(int[] t){
    int n= t.length;
    for (int i=0; i<n; i++){
        int indexMin=i;
        for (int j=i+1;j<n;j++){
            if (t[j]<t[indexMin]){
                indexMin=j;
            }
        }
        if (i!=indexMin){
            int temp=t[i];
            t[i]=t[indexMin];
            t[indexMin]=temp;
        }
    }
}
```

- ☐ $O(N \cdot \log_2(N))$
- ☐ $O(N^3)$
- ☐ $O(1)$
- ☐ $O(2^N)$
- ☐ $O(N)$
- ☐ $O(\log_2(N))$
- ☐ $O(N^2)$

Question 2 Si N est la taille du tableau t . Quelle est la complexité au pire cas, la plus précise, pour la fonction `f` ?

```
int f(int [] t){
    int j=t.length;
    return j;
}
```

- ☐ $O(2^N)$
- ☐ $O(\log_2(N))$
- ☐ $O(N)$
- ☐ $O(1)$
- ☐ $O(N^3)$
- ☐ $O(N^2)$
- ☐ $O(N \cdot \log_2(N))$

Question 3 Si N est la taille des données d'entrée, $O(2N^3 + 180N + 2000)$ est égal à:

- ☐ $O(N^2)$
- ☐ $O(N^3)$
- ☐ $O(\log_2(N))$
- ☐ $O(N \cdot \log_2(N))$
- ☐ $O(2^N)$
- ☐ $O(N)$
- ☐ $O(1)$

Question 4 Si N est la taille du tableau t . Quelle est la complexité au pire cas, la plus précise, pour la fonction `f` ?

```
void f(int [] t){
    int j=0;
    for(int i=0; i<t.length; i++){
        j=j;
    }
}
```

- ☐ $O(\log_2(N))$
- ☐ $O(N^2)$
- ☐ $O(N \cdot \log_2(N))$
- ☐ $O(2^N)$
- ☐ $O(1)$
- ☐ $O(N)$
- ☐ $O(N^3)$



Question 5 Si N est la taille du tableau t . Quelle est la complexité au pire cas, la plus précise, pour la fonction f ?

```
void f(int [] t){
    int j=0;
    for(int i=0; i<t.length; i++){
        j=j+t.length;
    }
}
```

- ☐ $O(2^N)$
- ☐ $O(\log_2(N))$
- ☐ $O(N^2)$
- ☐ $O(N^3)$
- ☐ $O(1)$
- ☐ $O(N)$
- ☐ $O(N \cdot \log_2(N))$

Question 6 Si N est la taille du tableau t . Quelle est la complexité au pire cas, la plus précise, pour la fonction f ?

```
void f(int [] t){
    int j=0;
    int len= t.length/2;
    for(int i=0; i < len; i++){
        j=j++;
    }
}
```

- ☐ $O(\log_2(N))$
- ☐ $O(N)$
- ☐ $O(N^3)$
- ☐ $O(1)$
- ☐ $O(N \cdot \log_2(N))$
- ☐ $O(N^2)$
- ☐ $O(2^N)$

Question 7 Si N est la taille des données d'entrée, dans quelle classe de complexité trouvera-t-on les fonctions les plus efficaces?

- ☐ $O(N)$
- ☐ $O(N \cdot \log_2(N))$
- ☐ $O(2^N)$
- ☐ $O(N^3)$
- ☐ $O(1)$
- ☐ $O(\log_2(N))$
- ☐ $O(N^2)$

Question 8 Quel est le pire cas pour la fonction f suivante:

```
int f(int [] t, int c){
    for(int i=0; i<t.length; i++){
        if(t[i]==c){
            return i;
        }
    }
    return -1;
}
```

- ☐ $t=\{2,2,2\}$ et $c=2$
- ☐ $t=\{2,1,3\}$ et $c=3$
- ☐ $t=\{1,2,3\}$ et $c=1$
- ☐ $t=\{3,2,1\}$ et $c=4$

L1 Informatique et Électronique – Première Session 2017-2018

SI1 : Algorithmique et complexité Expérimentale

2 heures - Documents autorisés

QCM (questions 1 à 8) : noircir les cases correspondant aux bonnes réponses sur le sujet.

Partie rédactionnelle (question 9 et 10) : à rédiger sur la copie anonymisée, faire les déroulages d'algorithmes sur le sujet.

Pour les algorithmes et programmes, on attend de la syntaxe Java. Le barème est donné à titre indicatif. **Il n'est nécessaire de dérouler un algorithme que lorsque cela est demandé explicitement.** Si vos algorithmes/programmes utilisent des fonctions vues en **Cours Magistral**, il n'est pas nécessaire de redonner leur code, donnez simplement leur signature/entête.

Question 9 (6pts) On souhaite programmer une fonction `void remplacer(int a, int b, int [] t)` qui remplace dans le tableau `t` toutes les occurrences de l'entier `a` par l'entier `b`. Par exemple, avec `t={1,4,1,3,10,1}`, `a=1` et `b=2`, le tableau `t` devient `t={2,4,2,3,10,2}`. On remarque que, même si `t` est trié, le tableau modifié par `remplacer` ne sera pas nécessairement trié.

1. Si `t` n'est pas trié :
 - (a) Quelle est la meilleure complexité, au pire cas, que l'on peut espérer pour `remplacer` ? Justifiez et expliquez à l'aide d'exemples illustrant le pire cas.
 - (b) Donnez le code de la fonction `remplacer`.
2. Si `t` est trié par ordre croissant :
 - (a) Quelle est la meilleure complexité, au pire cas, que l'on peut espérer pour `remplacer` ? Justifiez et expliquez à l'aide d'exemples illustrant le pire cas.
 - (b) Donnez le code de la fonction `remplacer`.
3. Si `t` est trié par ordre croissant et que chaque entier n'apparaît qu'une seule fois dans `t` :
 - (a) Quelle est la meilleure complexité, au pire cas, que l'on peut espérer pour `remplacer` ? Justifiez.
 - (b) Donnez le code de la fonction `remplacer`.

Question 10 (5pts) On propose la fonction `g` dont le code est donné au verso de cette feuille.

1. Proposez des valeurs pour les paramètres `t1` et `t2` pour une exécution au pire cas pour `g`.
2. Sur les feuilles suivantes, déroulez l'exécution de la fonction `g` sur deux de ces valeurs.
3. Sur les feuilles suivantes, déroulez l'exécution de la fonction `g` de façon abstraite sur deux de ces valeurs.
4. Déduisez-en la fonction $f(N)$ qui donne le nombre d'exécutions abstraites pour une valeur N qui est le minimum de la taille des tableaux `t1` et `t2`.
5. Donnez la complexité au pire cas de `g`, justifiez.

```
public static void g(int t1[], int t2 []){  
  
    int a= 0;  
  
    if (t1.length>t2.length)  
  
        a=t2.length;  
  
    else a=t1.length;  
  
    for (int i=0; i<a; i++){  
  
        int tmp= t1[i];  
  
        t1[i]=t1[i]+t2[i];  
  
        t2[i]=t2[i]+tmp;  }}
```

```
public static void g(int t1[], int t2 []){  
  
    int a= 0;  
  
    if (t1.length>t2.length)  
  
        a=t2.length;  
  
    else a=t1.length;  
  
    for (int i=0; i<a; i++){  
  
        int tmp= t1[i];  
  
        t1[i]=t1[i]+t2[i];  
  
        t2[i]=t2[i]+tmp;  }}
```

```
public static void g(int t1[], int t2 []){  
  
    int a= 0;  
  
    if (t1.length>t2.length)  
  
        a=t2.length;  
  
    else a=t1.length;  
  
    for (int i=0; i<a; i++){  
  
        int tmp= t1[i];  
  
        t1[i]=t1[i]+t2[i];  
  
        t2[i]=t2[i]+tmp; }}  
  
public static void g(int t1[], int t2 []){  
  
    int a= 0;  
  
    if (t1.length>t2.length)  
  
        a=t2.length;  
  
    else a=t1.length;  
  
    for (int i=0; i<a; i++){  
  
        int tmp= t1[i];  
  
        t1[i]=t1[i]+t2[i];  
  
        t2[i]=t2[i]+tmp; }}
```

```
public static void g(int t1[], int t2 []){  
  
    int a= 0;  
  
    if (t1.length>t2.length)  
  
        a=t2.length;  
  
    else a=t1.length;  
  
    for (int i=0; i<a; i++){  
  
        int tmp= t1[i];  
  
        t1[i]=t1[i]+t2[i];  
  
        t2[i]=t2[i]+tmp; }}  
  
}
```

```
public static void g(int t1[], int t2 []){  
  
    int a= 0;  
  
    if (t1.length>t2.length)  
  
        a=t2.length;  
  
    else a=t1.length;  
  
    for (int i=0; i<a; i++){  
  
        int tmp= t1[i];  
  
        t1[i]=t1[i]+t2[i];  
  
        t2[i]=t2[i]+tmp; }}  
  
}
```