Université Paris 7 – Licence d'informatique IF2 – 20 mai 2009

tous les documents sont interdits

Nom: _____

IF 2

QCM Final, Version: A

Carte d'étudiant: _____

Remplissez la table avec les lettres correspondant à vos réponses.

Questions	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Réponse(s)															

Questions	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
Réponse(s)															

Questions	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45
Réponse(s)															

Questions	46	47	48	49	50
Réponse(s)					

Bonne réponse=1pt; mauvaise réponse ou réponse incomplète =-0,5pt; pas de réponse=0pt. On rappelle qu'un arbre binaire parfait est un arbre dont tous les noeuds qui ne sont pas des feuilles ont exactement deux fils et toutes les branches (chemin de la racine à une feuille) ont la même longueur.

On rappelle que f est $\Theta(g)$ si et seulement si il existe m a et b tel que pour tout $n > m$ on a $ag(n) \leq f(n) \leq bg(n)$ (en informatique Θ est souvent noté $\mathcal{O}$). Dans la suite log *représente le logarithme en base 2.*

1. On considère la classe définie par : class A {static int i; void print(){System.out.println(i); }}
   L'exécution de A a=new A(); A b=new A(); a.i=5; b.i=10; a.print();
   (a) affiche 5
   (b) affiche 10
   (c) la classe A ne peut pas être compilée

2. On considère la boucle ($n > 0$): for(int i=0; i<n; i+=1)do f(i++); while(i<0); Trouver la bonne réponse:
   (a) le nombre d'appels de f est un $\Theta(n)$
   (b) le nombre d'appels de f est un $\Theta(n^2)$
   (c) le nombre d'appels de f est logarithmique ($\Theta(\log(n))$)
   (d) le nombre d'appels de f est un $\Theta(n\log(n))$
   (e) le nombre d'appels de f est exponentiel ($\Theta(2^n)$)

3. On considère le programme suivant:

```
static boolean estPresent(int v,int[]t, int l,int r){
    if(l>r) return false; int m=(l+r)/2;
    if(v==t[m])return true;
    if(v<t[m])return estPresent(v,t,l,m-1); else return estPresent(v,t,m+1,r);}
```

   En supposant que le tableau t est trié, ce programme retourne true si et seulement si v est présent dans t entre les indices l et r en (choisir la (ou les) bonne(s) réponse(s):
   (a) $\Theta(n)$ comparaisons dans le pire cas
   (b) $\Theta(\log(n))$ comparaisons dans le pire cas
   (c) $\Theta(n)$ comparaisons dans le meilleur cas

4. Pour un arbre binaire parfait de hauteur n, le nombre de noeuds est un:
   (a) 2^n
   (b) égal au nombre de feuilles d'un arbre parfait de hauteur $n+1$ moins 1
   (c) $n * \log(n)$

5. Trouver la ou les bonnes réponses:
   (a) $\Sigma_{i=0}^{i=n} i^3$ est un $\Theta(n^3)$
   (b) $\Sigma_{i=0}^{i=n} i^2$ est un $\Theta(n^3)$
   (c) $\Sigma_{i=0}^{i=n} i$ est un $\Theta(n^3)$

6. On considère le morceau de programme: tmp=t[0]; for(int i=1; i<t.length; i++)if(tmp<t[i])tmp=t[i];
   Choisir parmi les assertions suivantes celle(s) qui est(sont) vraies:
   (a) "tmp est égal au max du tableau t entre les indices 0 et $i-1$" est un invariant de la boucle
   (b) "tmp est égal au max du tableau t entre les indices 0 et $t.length$" est un invariant de la boucle
   (c) il n'y a pas d'invariant pour cette boucle

7. Le tri fusion trie un tableau de n éléments en:
   (a) $\Theta(n^2)$ comparaisons dans le meilleur cas
   (b) $\Theta(nlog(n))$ comparaisons dans le meilleur cas
   (c) $\Theta(n)$ comparaisons dans le meilleur cas

8. On considère la classe définie par : class A {int i=0; }. L'exécution de:
   A a,b; a.i=10; b=a; b.i=5; System.out.println(a.i);
   (a) affiche 10
   (b) affiche 5
   (c) provoque une erreur

9. On considère la boucle suivante ($n > 0$): `for(int i=1; i<n; i=i*10) f(i);` Parmi les assertions suivantes laquelle est FAUSSE:
   (a) le nombre d'appels de f est un Θ du nombre n
   (b) le nombre d'appels de f est un Θ du nombre de bits nécessaires pour représenter n en binaire
   (c) le nombre d'appels de f est un Θ du nombre de chiffres dans la représentation de n en base 10

10. Pour le morceau de programme: `if (a>b) m=a; else m=b;` et la pré-condition : a = A et b = B (où A et B sont des constantes positives), parmi les assertions suivantes laquelle n'est **pas** une post-condition:
   (a) m= $\max(A, B)$
   (b) $1 \neq 0$
   (c) m$\geq |A - B|$
   (d) $A > B$

11. On considère le programme suivant:
`static boolean cherche(int v, int[] t, int l, int r)`
`{for(int i=l;i<=r;i++)if(t[i]==v)return true; return false; }` Ce programme retourne true si et seulement si v est présent dans t entre les indices l et r en : (choisir la (ou les) bonne(s) réponse(s))
   (a) $\Theta(n)$ comparaisons dans le pire cas
   (b) $\Theta(\log(n))$ comparaisons dans le pire cas
   (c) $\Theta(n)$ comparaisons dans le meilleur cas

12. Pour la classe D définie comme suit:

```
class D {
public int x;
public D() {x=3; };
public D( int a){this(); x=x+a;};
public D( int a, int b){this(b); x= x-a;}}
```

qu'affichera le code suivant?

```
D a=new D(5,6);
System.out.println(a.x);
```

   (a) 1
   (b) 2
   (c) 3
   (d) 4

13. On considère le programme suivant:
`static void f(int n){if(n==0) return; f(n-1); System.out.print(n); f(n-1); }`
L'appel $f(4)$ affiche:
   (a) 121312141213121
   (b) 112112311211234
   (c) 432112113211211

14. On considère la classe définie par : `class A {int i; }`
L'exécution de: `A a=new A(); A b=new A(); a.i=8; b=a; b.i=5; System.out.println(a.i);`
   (a) affiche 5
   (b) affiche 8
   (c) provoque une erreur

15. On considère le programme suivant:
`static void f(int n){if(n<=0) return; f(n-4);f(n+1);}`
Le nombre d'appels de f pour $f(n)$ est un (on suppose $n > 0$):
   (a) $\Theta(n)$ dans le pire cas
   (b) $\Theta(\log(n))$ dans le pire cas
   (c) $\Theta(2^n)$ dans le pire cas
   (d) infini: le programme ne termine pas toujours

16. On considère la classe définie par : class A {int i; }. L'exécution de:
A a=new A();A b=new A(); a.i=10; b=a; b.i=5;
if(a==b)System.out.println("EGAL"); else System.out.println("PAS EGAL");

    (a) affiche EGAL

    (b) affiche PAS EGAL

    (c) provoque une erreur

17. On considère la classe définie par : class A {int i; void print(){System.out.println(i); }}
L'exécution de A a=new A(); A b=new A(); a.i=5; b.i=10; a.print();

    (a) affiche 5

    (b) affiche 10

    (c) la classe A ne peut pas être compilée

18. On considère le programme suivant:

```
static boolean estPresent(int v,int[]t, int l,int r){
    if(l>r) return false; int m=(l+r)/2;
    if(v==t[m])return true;
    return estPresent(v,t,l,m-1)|| estPresent(v,t,m+1,r);}
```

    En supposant que le tableau t est trié, ce programme retourne true si et seulement si v est présent dans t entre les indices l et r en (choisir la (ou les) bonne(s) réponse(s):

    (a) $\Theta(n)$ comparaisons dans le pire cas

    (b) $\Theta(\log(n))$ comparaisons dans le pire cas

    (c) $\Theta(n)$ comparaisons dans le meilleur cas

19. On définit la méthode permuter:

```
public static void permuter (String s1, String s2, int x1, int x2){
    String tmp1=s1; s1=s2; s2=tmp1; int tmp2=x1; x1=x2; x2=tmp2;
}
```

    Considérons: String a="bon"; String b="jour"; int c=3; int d =4; permuter(a,b,c,d);
    Quelles seront les valeurs de a,b,c,d après l'exécution de ce code?

    (a) "bon", "jour", 3, 4

    (b) "jour", "bon", 3, 4

    (c) "bon", "jour", 4, 3

    (d) "jour", "bon", 4, 3

20. Le tri par insertion trie un tableau de n éléments en:

    (a) $\Theta(n^2)$ comparaisons dans le pire cas

    (b) $\Theta(nlog(n))$ comparaisons dans le pire cas

    (c) $\Theta(n)$ comparaisons dans le pire cas

21. On considère le programme suivant:
static void f(int n){if(n<=0) return; if ((n%2)==0) f(n-1); else f(n+1); }
Le nombre d'appels de f pour $f(n)$ est un (on suppose $n > 0$):

    (a) $\Theta(n)$ dans le pire cas

    (b) $\Theta(\log(n))$ dans le pire cas

    (c) $\Theta(2^n)$ dans le pire cas

    (d) infini: le programme ne termine pas toujours

22. Trouver la ou les bonnes réponses:

    (a) $\Sigma_{i=0}^{i=n} i^3$ est un $\Theta(n^2)$

    (b) $\Sigma_{i=0}^{i=n} i^2$ est un $\Theta(n^2)$

    (c) $\Sigma_{i=0}^{i=n} i$ est un $\Theta(n^2)$

23. On considère la classe définie par : class A {int i=0; B b; }, la classe: B {int j=0;} et le code A a1=new A(); A a2=new A(); B b=new B();
Quelles expressions ont la valeur true:

    (a) (a1.i != b.j)

    (b) (a1.b == a2.b)

    (c) (a1 == a2)

24. On considère la boucle suivante ($n > 0$): for(int i=0; i<n; i+=1) for(int j=1; j<n; j=2*j) f(i); Trouver la bonne réponse:

    (a) le nombre d'appels de f est un $\Theta(n)$

    (b) le nombre d'appels de f est un $\Theta(n^2)$

    (c) le nombre d'appels de f est un $\Theta(\log(n))$

    (d) le nombre d'appels de f est un $\Theta(n\log(n))$

25. On considère le programme suivant:
static void f(int n){if(n==0) return; f(n-1); f(n-1); System.out.print(n); }
L'appel $f(4)$ affiche:

    (a) 121312141213121

    (b) 112112311211234

    (c) 432112113211211

26. On considère le programme suivant:
static void f(int n){if(n==0) return; System.out.print(n); f(n-1); f(n-1)}
L'appel $f(4)$ affiche:

    (a) 121312141213121

    (b) 112112311211234

    (c) 432112113211211

27. Trouver la ou les bonnes réponses:

    (a) $n^2/4 + \log(n)$ est un $\Theta(\log(n))$

    (b) $n\log(n)$ est un $\Theta(n)$

    (c) $n^2/4 + 8n^3$ est un $\Theta(n^3)$

28. On considère le programme suivant:
static void f(int n){if(n<=0) return; f(n-4);f(n-2);}
Le nombre d'appels de f pour $f(n)$ est un (on suppose $n > 0$):

    (a) $\Theta(n)$ dans le pire cas

    (b) $\Theta(\log(n))$ dans le pire cas

    (c) $\Theta(2^n)$ dans le pire cas

    (d) infini: le programme ne termine pas toujours

29. Le tri par insertion trie un tableau de n éléments en:

    (a) $\Theta(n^2)$ comparaisons dans le meilleur cas

    (b) $\Theta(nlog(n))$ comparaisons dans le meilleur cas

    (c) $\Theta(n)$ comparaisons dans le meilleur cas

30. Soit interface I {public void f(); } et le code suivant class A implements I {public void f(){System.out.println("bonjour"); }. Parmi les morceaux de code suivant lesquelles sont correctes (peuvent être compilés et ne provoquent pas d'erreur à l'exécution):

    (a) I i = new A(); i.f();

    (b) I i = new I(); i.f();

    (c) A a = new A(); I i=a; a=i; a.f();

31. On considère la classe définie par : `class A {int i; static void print(){System.out.println(i); }}`. L'exécution de:
`A a=new A(); A b=a; a.i=5; b.i=10; a.print();`
    (a) affiche 5
    (b) affiche 10
    (c) la classe A ne peut pas être compilée

32. Trouver la ou les bonnes réponses:
    (a) $n^2 + n$ est un $\Theta(n^2)$
    (b) $\log(n) * n^2$ est un $\Theta(n^2)$
    (c) $\log(n) * n$ est un $\Theta(n^2)$

33. On considère la boucle ($n > 0$): `for(int i=0; i<n; i+=3) f(i);` Trouver la bonne réponse:
    (a) le nombre d'appels de f est un $\Theta(n)$
    (b) le nombre d'appels de f est un $\Theta(\log(n))$
    (c) le nombre d'appels de f est un $\Theta(10^n)$

34. Soit la classe : `class A {public int i; }`. Le code suivant `A a=new A();Object o=a;o.i=10;`:
    (a) est correct (il peut être compilé et exécuté)
    (b) n'est pas correct (il ne peutpas être compilé et exécuté)

35. Pour un arbre binaire parfait de hauteur $\Theta(n)$, le nombre de noeuds est un:
    (a) $\Theta(2^n)$
    (b) $\Theta(n)$
    (c) $\Theta(\log(n))$

36. Soit la classe : `class A {public int i; }`. Le code suivant `A a=new A();Object o=a;((A)o).i=10;`:
    (a) est correct (il peut être compilé et exécuté)
    (b) n'est pas correct (il ne peut pas être compilé et exécuté)

37. On considère le programme suivant:
`static void f(int n){if(n<=1) return; f(n/2);f(n/2);}`
Le nombre d'appels de f pour $f(n)$ est un (on suppose $n > 0$):
    (a) $\Theta(n)$ dans le pire cas
    (b) $\Theta(n \log(n))$ dans le pire cas
    (c) $\Theta(2^n)$ dans le pire cas

38. Pour un arbre binaire parfait de hauteur n, le nombre de feuilles est:
    (a) 2^n
    (b) n
    (c) $\log(n)$

39. On considère la boucle ($n > 0$): `for(int i=0; i<n; i+=1)for(int j=n; j>0;j- -)f(i);` Trouver la bonne réponse:
    (a) le nombre d'appels de f est un $\Theta(n)$
    (b) le nombre d'appels de f est un $\Theta(n^2)$
    (c) le nombre d'appels de f est un $\Theta(\log(n))$
    (d) le nombre d'appel de f est un $\Theta(n \log(n))$
    (e) le nombre d'appels de f est exponentiel ($\Theta(2^n)$)

40. On considère la classe définie par : `class A {static int i; static void print(){System.out.println(i); }}`. L'exécution de:
`A a=new A(); A b=new A(); a.i=5; b.i=10; a.print();`
    (a) affiche 5
    (b) affiche 10
    (c) la classe A ne peut pas être compilée

41. Le tri rapide (quick-sort) trie un tableau de n éléments en:
    (a) $\Theta(n^2)$ comparaisons dans le pire cas
    (b) $\Theta(nlog(n))$ comparaisons dans pire cas
    (c) $\Theta(n)$ comparaisons dans le pire cas

42. Pour un arbre binaire parfait ayant $\Theta(n)$ feuilles, la hauteur est un:
    (a) $\Theta(2^n)$
    (b) $\Theta(\log(n))$
    (c) $\Theta(n)$

43. Quel est le résultat du morceau de code : `int i,j; i=10; j=i; j=5; System.out.println(i);`
    (a) il affiche `5`
    (b) il affiche `10`
    (c) il provoque une erreur

44. On considère la boucle ($n > 0$): `for(int i=0; i<n; i+=1) f(i);` Trouver la bonne réponse:
    En fonction du *nombre k de chiffres* dans la représentation de `n`:
    (a) le nombre d'appels de `f` est un $\Theta(k)$
    (b) le nombre d'appels de `f` est un $\Theta(\log(k))$
    (c) le nombre d'appels de `f` est un $\Theta(10^k)$

45. On considère la classe définie par : `class A {static int i; }`. L'exécution de:
    `A a=new A(); A b=new A(); a.i=10; b.i=5; System.out.println(a.i);`
    (a) affiche `5`
    (b) affiche `10`
    (c) provoque une erreur

46. On considère la boucle ($n > 0$): `for(int i=n; i>1; i/=10) f(i);` Trouver la bonne réponse:
    (a) le nombre d'appels de `f` est un $\Theta(n)$
    (b) le nombre d'appels de `f` est un $\Theta(n^2)$
    (c) le nombre d'appels de `f` est un $\Theta(\log(n))$
    (d) le nombre d'appels de `f` est un $\Theta(n\log(n))$

47. On considère la classe définie par : `class A {int i; }`. L'exécution de:
    `A a=new A(); A b=new A(); a.i=10; b.i=10;`
    `if(a==b)System.out.println("EGAL"); else System.out.println("PAS EGAL");`
    (a) affiche `EGAL`
    (b) affiche `PAS EGAL`
    (c) provoque une erreur

48. On considère le programme suivant:
    `static int f(int n){if(n<=1) return 0; else return f(n/2)+f(n/2); }`
    Le nombre d'appels de f pour $f(n)$ est un (on suppose $n > 0$):
    (a) $\Theta(n)$ dans le pire cas
    (b) $\Theta(\log(n))$ dans le pire cas
    (c) $\Theta(2^n)$ dans le pire cas

49. Le tri bulle trie un tableau de n éléments en:
    (a) $\Theta(log^2(n))$ comparaisons dans le pire cas
    (b) $\Theta(n)$ comparaisons dans le pire cas
    (c) $\Theta(n^2)$ comparaisons dans le pire cas

50. Trouver la ou les bonnes réponses:
    (a) $n^2 + n\log(n)$ est un $\Theta(\log(n))$
    (b) $n\log(n)$ est un $\Theta(n)$
    (c) $n^2\log(n) + n^3$ est un $\Theta(n^3)$

Answer Key for Exam A

Bonne réponse=1pt; mauvaise réponse ou réponse incomplète =-0,5pt; pas de réponse=0pt. On rappelle qu'un arbre binaire parfait est un arbre dont tous les noeuds qui ne sont pas des feuilles ont exactement deux fils et toutes les branches (chemin de la racine à une feuille) ont la même longueur.

On rappelle que f est $\Theta(g)$ si et seulement si il existe m a et b tel que pour tout $n > m$ on a $ag(n) \leq f(n) \leq bg(n)$ (en informatique Θ est souvent noté $\mathcal{O}$). Dans la suite log *représente le logarithme en base 2.*

1. On considère la classe définie par : class A {static int i; void print(){System.out.println(i); }}
   L'exécution de A a=new A(); A b=new A(); a.i=5; b.i=10; a.print();

   (a) affiche 5
   [(b)] affiche 10
   (c) la classe A ne peut pas être compilée

2. On considère la boucle ($n > 0$): for(int i=0; i<n; i+=1)do f(i++); while(i<0); Trouver la bonne réponse:

   [(a)] le nombre d'appels de f est un $\Theta(n)$
   (b) le nombre d'appels de f est un $\Theta(n^2)$
   (c) le nombre d'appels de f est logarithmique ($\Theta(\log(n))$)
   (d) le nombre d'appels de f est un $\Theta(n\log(n))$
   (e) le nombre d'appels de f est exponentiel ($\Theta(2^n)$)

3. On considère le programme suivant:

```
static boolean estPresent(int v,int[]t, int l,int r){
    if(l>r) return false; int m=(l+r)/2;
    if(v==t[m])return true;
    if(v<t[m])return estPresent(v,t,l,m-1); else return estPresent(v,t,m+1,r);}
```

   En supposant que le tableau t est trié, ce programme retourne true si et seulement si v est présent dans t entre les indices l et r en (choisir la (ou les) bonne(s) réponse(s):

   (a) $\Theta(n)$ comparaisons dans le pire cas
   [(b)] $\Theta(\log(n))$ comparaisons dans le pire cas
   (c) $\Theta(n)$ comparaisons dans le meilleur cas

4. Pour un arbre binaire parfait de hauteur n, le nombre de noeuds est un:

   (a) 2^n
   [(b)] égal au nombre de feuilles d'un arbre parfait de hauteur $n+1$ moins 1
   (c) $n * \log(n)$

5. Trouver la ou les bonnes réponses:

   (a) $\Sigma_{i=0}^{i=n} i^3$ est un $\Theta(n^3)$
   [(b)] $\Sigma_{i=0}^{i=n} i^2$ est un $\Theta(n^3)$
   (c) $\Sigma_{i=0}^{i=n} i$ est un $\Theta(n^3)$

6. On considère le morceau de programme: tmp=t[0]; for(int i=1; i<t.length; i++)if(tmp<t[i])tmp=t[i];
   Choisir parmi les assertions suivantes celle(s) qui est(sont) vraies:

   [(a)] "tmp est égal au max du tableau t entre les indices 0 et $i-1$" est un invariant de la boucle
   (b) "tmp est égal au max du tableau t entre les indices 0 et $t.length$" est un invariant de la boucle
   (c) il n'y a pas d'invariant pour cette boucle

7. Le tri fusion trie un tableau de n éléments en:

   (a) $\Theta(n^2)$ comparaisons dans le meilleur cas
   [(b)] $\Theta(nlog(n))$ comparaisons dans le meilleur cas
   (c) $\Theta(n)$ comparaisons dans le meilleur cas

8. On considère la classe définie par : class A {int i=0; }. L'exécution de:
   A a,b; a.i=10; b=a; b.i=5; System.out.println(a.i);

   (a) affiche 10
   (b) affiche 5
   [(c)] provoque une erreur

9. On considère la boucle suivante ($n > 0$): for(int i=1; i<n; i=i*10) f(i); Parmi les assertions suivantes laquelle est FAUSSE:

   [(a)] le nombre d'appels de f est un Θ du nombre n
   (b) le nombre d'appels de f est un Θ du nombre de bits nécessaires pour représenter n en binaire
   (c) le nombre d'appels de f est un Θ du nombre de chiffres dans la représentation de n en base 10

10. Pour le morceau de programme: `if (a>b) m=a; else m=b;` et la pré-condition : `a` $= A$ et `b` $= B$ (où A et B sont des constantes positives), parmi les assertions suivantes laquelle n'est **pas** une post-condition:

    (a) `m`$= \max(A, B)$

    (b) $1 \neq 0$

    (c) `m`$\geq |A - B|$

    [(d)] $A > B$

11. On considère le programme suivant:
    `static boolean cherche(int v, int[] t, int l, int r)`
    `{for(int i=l;i<=r;i++)if(t[i]==v)return true; return false; }` Ce programme retourne `true` si et seulement si `v` est présent dans `t` entre les indices `l` et `r` en : (choisir la (ou les) bonne(s) réponse(s))

    [(a)] $\Theta(n)$ comparaisons dans le pire cas

    (b) $\Theta(\log(n))$ comparaisons dans le pire cas

    (c) $\Theta(n)$ comparaisons dans le meilleur cas

12. Pour la classe D définie comme suit:

```
class D {
public int x;
public D() {x=3; };
public D( int a){this(); x=x+a;};
public D( int a, int b){this(b); x= x-a;}}
```

    qu'affichera le code suivant?

```
D a=new D(5,6);
System.out.println(a.x);
```

    (a) 1

    (b) 2

    (c) 3

    [(d)] 4

13. On considère le programme suivant:
    `static void f(int n){if(n==0) return; f(n-1); System.out.print(n); f(n-1); }`
    L'appel $f(4)$ affiche:

    [(a)] 121312141213121

    (b) 112112311211234

    (c) 432112113211211

14. On considère la classe définie par : `class A {int i; }`
    L'exécution de: `A a=new A(); A b=new A(); a.i=8; b=a; b.i=5; System.out.println(a.i);`

    [(a)] affiche 5

    (b) affiche 8

    (c) provoque une erreur

15. On considère le programme suivant:
    `static void f(int n){if(n<=0) return; f(n-4);f(n+1);}`
    Le nombre d'appels de f pour $f(n)$ est un (on suppose $n > 0$):

    (a) $\Theta(n)$ dans le pire cas

    (b) $\Theta(\log(n))$ dans le pire cas

    (c) $\Theta(2^n)$ dans le pire cas

    [(d)] infini: le programme ne termine pas toujours

16. On considère la classe définie par : `class A {int i; }`. L'exécution de:
    `A a=new A();A b=new A(); a.i=10; b=a; b.i=5;`
    `if(a==b)System.out.println("EGAL"); else System.out.println("PAS EGAL");`

    [(a)] affiche EGAL

    (b) affiche PAS EGAL

    (c) provoque une erreur

17. On considère la classe définie par : class A {int i; void print(){System.out.println(i); }}
L'exécution de A a=new A(); A b=new A(); a.i=5; b.i=10; a.print();

[(a)] affiche 5
(b) affiche 10
(c) la classe A ne peut pas être compilée

18. On considère le programme suivant:

```
static boolean estPresent(int v,int[]t, int l,int r){
    if(l>r) return false; int m=(l+r)/2;
    if(v==t[m])return true;
    return estPresent(v,t,l,m-1)|| estPresent(v,t,m+1,r);}
```

En supposant que le tableau t est trié, ce programme retourne true si et seulement si v est présent dans t entre les indices l et r en (choisir la (ou les) bonne(s) réponse(s):

[(a)] $\Theta(n)$ comparaisons dans le pire cas
(b) $\Theta(\log(n))$ comparaisons dans le pire cas
(c) $\Theta(n)$ comparaisons dans le meilleur cas

19. On définit la méthode permuter:

```
public static void permuter (String s1, String s2, int x1, int x2){
    String tmp1=s1; s1=s2; s2=tmp1; int tmp2=x1; x1=x2; x2=tmp2;
}
```

Considérons: String a="bon"; String b="jour"; int c=3; int d =4; permuter(a,b,c,d);
Quelles seront les valeurs de a,b,c,d après l'exécution de ce code?

[(a)] "bon", "jour", 3, 4
(b) "jour", "bon", 3, 4
(c) "bon", "jour", 4, 3
(d) "jour", "bon", 4, 3

20. Le tri par insertion trie un tableau de n éléments en:

[(a)] $\Theta(n^2)$ comparaisons dans le pire cas
(b) $\Theta(nlog(n))$ comparaisons dans le pire cas
(c) $\Theta(n)$ comparaisons dans le pire cas

21. On considère le programme suivant:
static void f(int n){if(n<=0) return; if ((n%2)==0) f(n-1); else f(n+1); }
Le nombre d'appels de f pour $f(n)$ est un (on suppose $n > 0$):

(a) $\Theta(n)$ dans le pire cas
(b) $\Theta(\log(n))$ dans le pire cas
(c) $\Theta(2^n)$ dans le pire cas
[(d)] infini: le programme ne termine pas toujours

22. Trouver la ou les bonnes réponses:

(a) $\Sigma_{i=0}^{i=n} i^3$ est un $\Theta(n^2)$
(b) $\Sigma_{i=0}^{i=n} i^2$ est un $\Theta(n^2)$
[(c)] $\Sigma_{i=0}^{i=n} i$ est un $\Theta(n^2)$

23. On considère la classe définie par : class A {int i=0; B b; }, la classe: B {int j=0;} et le code A a1=new A(); A a2=new A(); B b=new B();
Quelles expressions ont la valeur true:

(a) (a1.i != b.j)
[(b)] (a1.b == a2.b)
(c) (a1 == a2)

24. On considère la boucle suivante ($n > 0$): `for(int i=0; i<n; i+=1) for(int j=1; j<n; j=2*j) f(i);` Trouver la bonne réponse:

    (a) le nombre d'appels de f est un $\Theta(n)$

    (b) le nombre d'appels de f est un $\Theta(n^2)$

    (c) le nombre d'appels de f est un $\Theta(\log(n))$

    [(d)] le nombre d'appels de f est un $\Theta(n\log(n))$

25. On considère le programme suivant:
    `static void f(int n){if(n==0) return; f(n-1); f(n-1); System.out.print(n); }`
    L'appel $f(4)$ affiche:

    (a) 121312141213121

    [(b)] 112112311211234

    (c) 432112113211211

26. On considère le programme suivant:
    `static void f(int n){if(n==0) return; System.out.print(n); f(n-1); f(n-1)}`
    L'appel $f(4)$ affiche:

    (a) 121312141213121

    (b) 112112311211234

    [(c)] 432112113211211

27. Trouver la ou les bonnes réponses:

    (a) $n^2/4 + \log(n)$ est un $\Theta(\log(n))$

    (b) $n\log(n)$ est un $\Theta(n)$

    [(c)] $n^2/4 + 8n^3$ est un $\Theta(n^3)$

28. On considère le programme suivant:
    `static void f(int n){if(n<=0) return; f(n-4);f(n-2);}`
    Le nombre d'appels de f pour $f(n)$ est un (on suppose $n > 0$):

    (a) $\Theta(n)$ dans le pire cas

    (b) $\Theta(\log(n))$ dans le pire cas

    [(c)] $\Theta(2^n)$ dans le pire cas

    (d) infini: le programme ne termine pas toujours

29. Le tri par insertion trie un tableau de n éléments en:

    (a) $\Theta(n^2)$ comparaisons dans le meilleur cas

    (b) $\Theta(nlog(n))$ comparaisons dans le meilleur cas

    [(c)] $\Theta(n)$ comparaisons dans le meilleur cas

30. Soit `interface I {public void f();` } et le code suivant `class A implements I {public void f(){System.out.println("bonjour");` }. Parmi les morceaux de code suivant lesquelles sont correctes (peuvent être compilés et ne provoquent pas d'erreur à l'exécution):

    [(a)] `I i = new A(); i.f();`

    (b) `I i = new I(); i.f();`

    (c) `A a = new A(); I i=a; a=i; a.f();`

31. On considère la classe définie par : `class A {int i; static void print(){System.out.println(i); }}`. L'exécution de:
    `A a=new A(); A b=a; a.i=5; b.i=10; a.print();`

    (a) affiche 5

    (b) affiche 10

    [(c)] la classe A ne peut pas être compilée

32. Trouver la ou les bonnes réponses:

    [(a)] $n^2 + n$ est un $\Theta(n^2)$

    (b) $\log(n) * n^2$ est un $\Theta(n^2)$

    (c) $\log(n) * n$ est un $\Theta(n^2)$

33. On considère la boucle ($n > 0$): `for(int i=0; i<n; i+=3) f(i);` Trouver la bonne réponse:

- [x] (a) le nombre d'appels de f est un $\Theta(n)$
- [ ] (b) le nombre d'appels de f est un $\Theta(\log(n))$
- [ ] (c) le nombre d'appels de f est un $\Theta(10^n)$

34. Soit la classe : `class A {public int i; }`. Le code suivant `A a=new A();Object o=a;o.i=10;`:

- [ ] (a) est correct (il peut être compilé et exécuté)
- [x] (b) n'est pas correct (il ne peutpas être compilé et exécuté)

35. Pour un arbre binaire parfait de hauteur $\Theta(n)$, le nombre de noeuds est un:

- [x] (a) $\Theta(2^n)$
- [ ] (b) $\Theta(n)$
- [ ] (c) $\Theta(\log(n))$

36. Soit la classe : `class A {public int i; }`. Le code suivant `A a=new A();Object o=a;((A)o).i=10;`:

- [x] (a) est correct (il peut être compilé et exécuté)
- [ ] (b) n'est pas correct (il ne peut pas être compilé et exécuté)

37. On considère le programme suivant:
`static void f(int n){if(n<=1) return; f(n/2);f(n/2);}`
Le nombre d'appels de f pour $f(n)$ est un (on suppose $n > 0$):

- [x] (a) $\Theta(n)$ dans le pire cas
- [ ] (b) $\Theta(n\log(n))$ dans le pire cas
- [ ] (c) $\Theta(2^n)$ dans le pire cas

38. Pour un arbre binaire parfait de hauteur n, le nombre de feuilles est:

- [x] (a) 2^n
- [ ] (b) n
- [ ] (c) $\log(n)$

39. On considère la boucle ($n > 0$): `for(int i=0; i<n; i+=1)for(int j=n; j>0;j- -)f(i);` Trouver la bonne réponse:

- [ ] (a) le nombre d'appels de f est un $\Theta(n)$
- [x] (b) le nombre d'appels de f est un $\Theta(n^2)$
- [ ] (c) le nombre d'appels de f est un $\Theta(\log(n))$
- [ ] (d) le nombre d'appel de f est un $\Theta(n\log(n))$
- [ ] (e) le nombre d'appels de f est exponentiel ($\Theta(2^n)$)

40. On considère la classe définie par : `class A {static int i; static void print(){System.out.println(i); }}`. L'exécution de:
`A a=new A(); A b=new A(); a.i=5; b.i=10; a.print();`

- [ ] (a) affiche 5
- [x] (b) affiche 10
- [ ] (c) la classe A ne peut pas être compilée

41. Le tri rapide (quick-sort) trie un tableau de n éléments en:

- [x] (a) $\Theta(n^2)$ comparaisons dans le pire cas
- [ ] (b) $\Theta(nlog(n))$ comparaisons dans pire cas
- [ ] (c) $\Theta(n)$ comparaisons dans le pire cas

42. Pour un arbre binaire parfait ayant $\Theta(n)$ feuilles, la hauteur est un:

- [ ] (a) $\Theta(2^n)$
- [x] (b) $\Theta(\log(n))$
- [ ] (c) $\Theta(n)$

43. Quel est le résultat du morceau de code : `int i,j; i=10; j=i; j=5; System.out.println(i);`

(a) il affiche 5

[(b)] il affiche 10

(c) il provoque une erreur

44. On considère la boucle ($n > 0$): `for(int i=0; i<n; i+=1) f(i);` Trouver la bonne réponse:
En fonction du *nombre k de chiffres* dans la représentation de `n`:

(a) le nombre d'appels de `f` est un $\Theta(k)$

(b) le nombre d'appels de `f` est un $\Theta(\log(k))$

[(c)] le nombre d'appels de `f` est un $\Theta(10^k)$

45. On considère la classe définie par : `class A {static int i; }`. L'exécution de:
`A a=new A(); A b=new A(); a.i=10; b.i=5; System.out.println(a.i);`

[(a)] affiche 5

(b) affiche 10

(c) provoque une erreur

46. On considère la boucle ($n > 0$): `for(int i=n; i>1; i/=10) f(i);` Trouver la bonne réponse:

(a) le nombre d'appels de `f` est un $\Theta(n)$

(b) le nombre d'appels de `f` est un $\Theta(n^2)$

[(c)] le nombre d'appels de `f` est un $\Theta(\log(n))$

(d) le nombre d'appels de `f` est un $\Theta(n\log(n))$

47. On considère la classe définie par : `class A {int i; }`. L'exécution de:
`A a=new A(); A b=new A(); a.i=10; b.i=10;`
`if(a==b)System.out.println("EGAL"); else System.out.println("PAS EGAL");`

(a) affiche EGAL

[(b)] affiche PAS EGAL

(c) provoque une erreur

48. On considère le programme suivant:
`static int f(int n){if(n<=1) return 0; else return f(n/2)+f(n/2); }`
Le nombre d'appels de f pour $f(n)$ est un (on suppose $n > 0$):

(a) $\Theta(n)$ dans le pire cas

[(b)] $\Theta(\log(n))$ dans le pire cas

(c) $\Theta(2^n)$ dans le pire cas

49. Le tri bulle trie un tableau de n éléments en:

(a) $\Theta(log^2(n))$ comparaisons dans le pire cas

(b) $\Theta(n)$ comparaisons dans le pire cas

[(c)] $\Theta(n^2)$ comparaisons dans le pire cas

50. Trouver la ou les bonnes réponses:

(a) $n^2 + n\log(n)$ est un $\Theta(\log(n))$

(b) $n\log(n)$ est un $\Theta(n)$

[(c)] $n^2\log(n) + n^3$ est un $\Theta(n^3)$