Solution Série 1 :

Exercice 1 :

Algorithme A1 :

Début

Lire(n) ; 1u

premier = vrai ; 1u

i = 2 ; 1u

tant que (i <= n-1) et (premier=vrai) faire 3u

si (n mod i = 0) alors premier = faux //2u pour la cond , 1u alors

sinon i = i+1 ; 2u

fsi ; == 4 u

fait ; = 7*nb_it = 7*(n-2)u

ecrire(''le nombre n'est premier : '', premier) ; 1u

fin. = **A1(n)= 4+7*(n-2)+ 3=7*n-7 = O(n).**

Pire cas : n : est premier

Algorithme A2 :

Début

Lire(n) ; 1u

premier = vrai ; 1u

i = 2 ; 1u

tant que (i <= n div 2) et (premier=vrai) faire 4u

si (n mod i = 0) alors premier = faux //2u pour la cond , 1u alors

sinon i = i+1 ; 2u

fsi ; == 4 u

fait ; = 7*nb_it = 8*(n/2)u

ecrire(''le nombre n'est premier : '', premier) ; 1u

fin. = **A2(n)=4+8*(n/2)+4=4*n+12=O(n)**

Algorithme A3 :

Début

Lire(n) ; 1u

Si (n<>2) alors si (n mod 2=0) alors premier=faux ;

//1u + 2u (alors) ou 1u+ 2u+7*(n-3)/2+3

Sinon premier = vrai ; 1u

i = 3 ; 1u

tant que (i <= n-2) et (premier=vrai) faire 3u

si (n mod i = 0) alors premier = faux //2u pour la cond , 1u alors

sinon i = i+2 ; 2u

fsi ; == 4 u
fait ; = 7*nb_it = 7*(n-3)/2
fsi ; = 1u+ 2u+7*(n-3)/2+3
ecrire(''le nombre n'est premier : '', premier) ; 1u
fin. = **A3(n)= 2u+7*(n-3)/2+3+1=7/2 n+6-21/2= 7/2 n -9/2= O(n)**

Algorithme A4 :

Début

Lire(n) ; 1u
Si (n<>2) alors si (n mod 2=0) alors premier=faux ;
//1u + 2u (alors) ou 1u+ 2u+8(n/4-1/2)+4
Sinon premier = vrai ; 1u
i = 3 ; 1u
tant que (i <= n div 2) et (premier=vrai) faire 4u
si (n mod i = 0) alors premier = faux //2u pour la cond , 1u alors
sinon i = i+2 ; 2u
fsi ; == 4 u
fait ; = 7*nb_it = 8*(n/2+2-3)/2=8(n/4-1/2)
fsi ; = 1u+ 2u+8(n/4-1/2)+4
ecrire(''le nombre n'est premier : '', premier) ; 1u
fin. = **A4(n)=2 n- 4= O(n).**

Algorithme A5 :
Début
Lire(n) ; 1u
premier = vrai ; 1u
i = 2 ; 1u
tant que (i <= sqrt(n))) et (premier=vrai) faire 3u
si (n mod i = 0) alors premier = faux //2u pour la cond , 1u alors
sinon i = i+1 ; 2u
fsi ; == 4 u
fait ; = 7*nb_it = 7*(sqrt(n)-1)u
ecrire(''le nombre n'est premier : '', premier) ; 1u
fin. = $A5(n) = 4 + 7 * \left(\sqrt{n} - 1\right) + 3 = 7 * \sqrt{n}$= **O(**$\sqrt{n}$)

Algorithme A6 :
Lire(n) ; 1u
Si (n<>2) alors si (n mod 2=0) alors premier=faux ;
//1u + 2u (alors) ou 1u+ 2u+8(n/4-1/2)+4
Sinon premier = vrai ; 1u
i = 3 ; 1u

tant que (i <= sqrt(n)) et (premier=vrai) faire 4u
si (n mod i = 0) alors premier = faux //2u pour la cond , 1u alors
sinon i = i+2 ; 2u
fsi ; == 4 u
fait ; = 7*nb_it = 8*(sqrt(n)+2-3)/2=4*sqrt(n)-4
fsi ; = 1u+ 2u+4*sqrt(n)-4
ecrire(''le nombre n'est premier : '', premier) ; 1u

fin. = **A6(n)= 4*$\sqrt{n}$= O($\sqrt{n}$)**

Exemple n=990181

	L'expression	*Complexité*	*N=990181*
A1	**7*n-7**	*O(n)*	*6931260*
A2	**4*n+12**	*O(n)*	*3960736*
A3	**7/2 n -9/2**	*O(n)*	*3465629*
A4	**2 n- 4**	*O(n)*	*1980358*
A5	*7*$\sqrt{n}$*	**O($\sqrt{n}$)**	*6965*
A6	**4*$\sqrt{n}$**	**O($\sqrt{n}$)**	*3980*

Exercice 2 :

1 heure = 3600s = 3.6 10^3s

1 jour = 86400s = 8.64 10^4s

1 semaine = 604800s ≈ 6.05 10^5s

1 mois= 2 592 000s ≈ 2.59 10^6s

1 année =31 536 000 ≈ 3.15 10^7s

1 siècle =3 153 600 000≈3.15 10^9s

1 millénaire = 31 536 000 000 ≈ 3.15 10^{10}s

Algorithme et Complexité Temporelle	Temps					
	1 milliseconde = 10^{-3}s			1 microseconde= 10^{-6}s		
	N=10	N=100	N=1000	N=10	N=100	N=1000
A0=O(1)	10^{-3}	10^{-3}	10^{-3}	10^{-6}	10^{-6}	10^{-6}
A1=O($\log_2$ n)	3,32*10^{-3}	6,64*10^{-3}	9,96*10^{-3}	3,32*10^{-6}	6,64*10^{-6}	9,96*10^{-6}
A2=O($\sqrt{n}$)	3,16*10^{-3}	0,01	3,16*10^{-2}	3,16*10^{-6}	10^{-4}	3,16*10^{-5}
A3=O(n)	0,01	0,1	1	10^{-5}	10^{-4}	10^{-3}
A4=O(n^2)	0,1	10	1000	10^{-4}	0,01	1
A5=O(n^3)	1	1000	10^6	10^{-3}	1	1000
A6=(2^n)	1,024	3,2*10^{16} millénaire	3,2*10^{276} millénaire	1,02*10^{-3}	3,2*10^{13} millénaire	3,2*10^{273} millénaire

1000 s = 16mn40s

10^6s= 11j 13h46mn40s.

Exercice 3 :

Additions de m matrices carrées :

(((A1+A2)+A3)+ ……)+Am)

A1+A2=A1,2

A1,2+A3=A1,2,3 ……

A1,…..,m-1+Am=A

Complexité totale=Nombre d'additions de matrices deux à deux*complexité de l'addition de deux matrice.

Complexité totale = $(m-1)*n^2=m*n^2-n^2=O(m*n^2)$.

Exercice 4 :

$T1(n)= 400*n = O(n)$

$T2(n)= n^2 =O(n^2)$.

$O(n) < O(n^2)$. $T1(n) < T2(n)$.

$n^2 < 400*n \Rightarrow n < 400$.

Si $n <400$ $T2(n) < T1(n)$

$n < 400 \Rightarrow n*n <400*n \Rightarrow n^2 < 400 n$

Sinon ($n >=400$) $T1(n) < T2(n)$.

Exercice 5 :

∀

$O(1)< O(Log n) <O(n)<O(nlog n)<O(n^2)<O(n^2log n)<O(n^3)< O(2^n)<O(3^n)<O(n !)$

N ?

On a :

Log n >=1 ∀ n >= base du logarithme, n>10.

∀ n >0, log n < n nlog n <n^2 n^2logn<n^3

Puisque lorsque n>10, log n >1 nlog n > n n^2log n >n^2.

$O(n^3) <O(2^n)$???

$N^3 <2^n$ 3 log n < n log 2 log n/n < log 2 /3 log n /n < 1/10

Logn / n = 1/10 lorsque n=10.

Puisque il s'agit d'une fonction monotone décroissante alors : pour n > 10.

Log n /n < 1/10.

$A^{\log n}= n^{\log a}$

$2^n < 3^n$ pour tout n >0.

$3^n < n!$ $\log(3^n) < \log(n!)$ $n \log 3 < \log(1*2*3****N)$ $n \log3 < \sum \log i$ $n \log 3 < \log1+\log2+\log3+\ldots.. \log n < \log n + \log n+ \ldots. + \log n$

$n \log 3 < n \log n$ $n > 3$

$3^n < n!$ (par l'exemple) pour tout $n > 7$.

$N \log 3 < n \log n$ $3^n < n^n$.

Exercice 6 :

$T1(n)= 3 n\log n + \log n = O(n \log n)$

$3 n \log n + \log n < 3 n \log n+ n \log n$ $T1(n)< 4*n \log n.$, $c=4$, $n_0=0$.

Puisque $\log n < n \log n$ pour tout $n > 0$

$T2(n) = 2^n + n^3 + 25 < C* 2^n$

$T2(n)= O(2^n)$.

$2^n + n^3 + 25 < 2^n + 2^n + 2^n= 3*2^n$, $c=3$, $n_0= 10$

$N^3 < 2^n$, pour tout $n > 10$

$25 < 2^n$ $\log_2 25 < n > 4$.

$T3(n, k)= k +n$, $k <= n$.

$T3(n, k)= O(n)$.

$K < =n$ $k+n < = 2* n$, $c=2$. $n_0=0$.

$T4(n, k)= k*n$, $k <=n$.

$K<=n$ $k*n <= n*n$ $O(n^2)$.

Exercice 9 :

$2^{n+1}= 2*2^n =O(2^n)$.

$2^{2n}= (2^2)^n= (2^n)^2= 2^n * 2^n \neq c* 2^n$

$2^{2n}=O(4^n)$.

Exercice 7 :

$Max(f(n), g(n))= \Theta(f(n)+g(n))$.

$C1.(f(n)+g(n)) < Max(f(n), g(n))< C2*(f(n)+g(n))$

F, g deux fonctions positives.

Si $F(n)<g(n)$ $max(f(n), g(n)) = g(n)$

$F(n)+g(n) < 2* g(n)$.

Si $g(n) < f(n)$ $max(f(n), g(n))=f(n)$.

$F(n)+g(n) < 2*F(n)$.

A partir des deux situations on a :

$F(n)+g(n)< 2* Max(f(n), g(n))$.   ½ $(f(n)+g(n)< Max(f(n)+g(n))$; $c1=$ ½.

- Si $Max(f(n), g(n)) = f(n)$ $max(f((n)+g(n)< max(f(n), g(n))+g(n)$, g est une fonction positive.
- Si $max(f(n), g(n))=g(n)$ $max(f(n)+g(n) < max(f(n)+g(n) +f(n)$, f est une fonction positive.

Dans les deux cas, on a :

$max(f(n)+g(n) < f(n)+g(n)$; $c2=1$.

****$C1*max(f(n), g(n)) < F(n)+g(n) < C2* max(f(n), g(n))$???.

Exercice 8 :

$(n + a)^b= \Theta (n^b)$, a et b deux nombres réels, $b >0$, n entier positif.

$n - |a| < n + a < n + |a|$.

on a : $n > |a|$, pour que la dominance puisse être exprimée en fonction de n et non en fonction de a.

$n > |a|$ $n + |a| < 2* n$

donc, $n + a < 2* n$, si on applique la puissance en b on obtient :

$(n + a)^b < 2^b * n^b$; $c2= 2^b$, $n_0 =|a|$.

Si $n > |a|$ $|a| < n$

$n - |a| < n + a$ $0 < n + a$

?*n < n+ a.

Si n > 2* |a|, ½ n <n +a.

½ n < n + a, si on applique la puissance en b on obtient :

$(½)^b n^b < (n+a)^b$, $c1= (½)^b$, $n_0= 2* |a|$

Si n > 4*|a| |a| < ¼ n ¾ n < n+ a.

Si n > 10* |a| |a| < 1/10 n 9/10 n < n+ a.

Exercice 10 :

(a) $5n^2-6n=\Theta(n^2)$

$c1 * n^2 <= 5 n^2 - 6 n <= c2* n^2$, pour tout $n >= n_0$.

$5n^2 - 6n > 0$ $5 n^2 > 6*n$ $n > 6/5$. $n_0= 6/5$.

$5n^2 - 6n <= 5*n^2$, c2=5.

$n^2 < 4* n^2 < 5 n^2 - 6 n$, c1= 4.

$n^2 > 6*n$ n > 6. , $n_0=6$

pour c1= 4, $n_0=6$, $4* n^2 < 5 n^2 - 6 n < 5 * n^2$

pour c1=1, $n_0=6/5$. $1* n^2 < 5 n^2 - 6 n < 5* n^2$.

(b) $n ! = O(n^n)$

n != 1*2*3* …*(n-1)*n

pour tout n >1 :

1*2*3* …*(n-1)*n <= n*n*n*…*n*n $n ! < n^n$, c=1, $n_0=1$.

(c) $2 n^2+ n\log(n)= \Theta(n^2)$

$C1* n^2 < 2n^2+ n\log n < c2*n^2$, avec c1 et c2 >0, $n_0 >= 0$.

$2n^2+ n\log n > 2n^2$. Pour tout n >0, c1=2.

Pour tout n>0, log n < n $n\log n < n^2$

Donc $2n^2+ n \log n < 3 n^2$, avec c2= 3, $n_0= 0$

(d) $\sum_{i=0}^{n} i^2 = \Theta(n^3)$

$\sum_{i=0}^{n} i^2$ = n(n+1)(2n+1)/6 = [(n²+n)(2n+1)]/6 = (2n³+3n²+n)/6 = 1/3 n³+ ½ n²+1/6 n

$$\frac{n^3}{3} \le \sum_{i=0}^{n} i^2 = \frac{n(n+1)(2n+1)}{6} \le n^3 \quad \forall n \ge 1 \text{ , c1= 1/3, C2=1.}$$

(e) $\sum_{i=0}^{n} 3 \sum_{i=0}^{n} i^3 = \Theta(n^4)$

$\sum_{i=0}^{n} i^3$ = [n(n+1)/2]²= n²(n+1)²/4 = n²(n²+ 2n + 1)/4 = ¼ n⁴+ ½ n³ + ¼ n²

$$\frac{n^4}{4} \le \sum_{i=0}^{n} i^3 = \left(\frac{n(n+1)}{2}\right)^2 \le \frac{3}{4} n^4 < n^4 \quad \forall n \ge 1 \text{ , c1= ¼, C2= 1.}$$

(f) $n^{2^n} + 6*2^n = \Theta(n^{2^n})$

$C_1 n^{2^n} \le n^{2^n} + 6*2^n \le C_2 n^{2^n}$

$$C_1 \le 1 + \frac{6*2^n}{n^{2^n}} \le C_2$$

$$\forall n \ge 3 \quad 0 \le \frac{6*2^n}{n^{2^n}} \le 1 \quad \text{avec } C_1=1,\ C_2=2 \text{ et } n_0=3.$$

Lim x   infini F(x)/g(x) = infini / infini   non déterminisme

Règle de l'hopitale :

Si F(x) > g(x) (F(x) domine g(x)) , la limite = la limite (f(x)) = infini

Si G(x) > F(x) , dominance est pour g(x), la limite = la limite (1/g(x))= 0.

Exercice 11 :

A1 :

Pour j=1 à N faire …… fait ;

Nombre d'itérations de la première boucle = N.

Pour j=1 à i Faire …… fait ;

Borne inf =1

Borne sup=i+1

Pas = 1 = nombre d'itérations = i = 1*2*3* ….. *N = n !.

F(A1)= n+n != O(n !).

A2 :

Boucle interne

Pour j=1 à N faire …. Fait ;

Nombre d'itérations = N.

Boucle externe :

Borne inf = 1

Borne sup = 2*N

I > N : condition d'arrêt de la boucle

Pas n'est pas fixe et n'est pas constant.

1^{er} it : I=1, = i=2*i=2

2eme it : I=2, i=2*i=4

3eme it : I=4 i=8

4eme it: I=8, i = 16.

….

K-1 eme it : I=N/2 i=2*i=N

Keme it : I=N i= 2*i = 2*N.

$1*2^k = N$ k qui reflète le nombre d'itérations = $\log_2 n$

Donc, on a deux boucles imbriquées indépendantes le nombre d'itérations totale c'est le produit :

F(A2)= log n * n = O(n log n).

A3 :

Deux boucles séquentielles :

Première boucle :

Pour j=1 à N faire … fait ; // calcul de i= n !

Nombre d'itérations = N.

Deuxième boucle :

Borne inf = 1

Borne sup = i = n !

Pas : en puissance de 2.

$1*2^k= n !$ $k = \log_2 (n !)$

Log (n !) = log(1*2*3….*N)= log(1)+log(2)+ …. Log (n) <= n* log n

K<= n* log n O(n log n).

F(A3) = n+ n log n = O(n log n)

<u>A4 :</u>

i=2,

Borne inf = 2

Borne sup > N (N^2)

Pas : calculer de la forme : i^2

1^{er} it : i=2 $i=2^2$.

2eme it : $i=2^2$ $i=(2^2)^2$ = 2^2^2

3eme it : i=16 $i=16^2=((2^2)^2)^2$= 2^2^3

Keme it : i=(2)^2^(k-1) i=(2)^2^k

Le nombre d'itérations correspond au nombre puissance de 2 calculés

Pour sortir de la boucle : (2)^2^k = N $2^k = \log_2 N$ k == log log N

F(A4)= O(log log N).

<u>A5 :</u>

I=1 ; j=1 ;

Tant que (i <=N)

Faire si i<N alors i=i+1 ;

    Sinon j=j+1 ; i=j ;

    Fsi ;

Fait ;

Fin.

J=1, i=1 bloc alors donc i=2

J=1, i=2 bloc alors donc i=3

J=1, i=3 bloc alors donc i=4

……

J=1, i=n-1 bloc alors i=n

J=1, i=n bloc sinon j=2 ; i=2

J=2, i=2 bloc alors i=3

…..

J=2, i=n bloc sinon j=3, i=3

J=3, i=3 bloc alors i=4

…..

J=3, i=n bloc sinon j=4, i=4.

……….

J=n-1,i=n-1 bloc alors i=n

J=n-1, i=n bloc sinon j=n, i=n

J=n, i=n bloc sinon j=n+1, i=n+1

La fin de la boucle

Nombre d'itérations= n+(n-1)+(n-2)+…..+2+1= n(n+1)/2= $O(n^2)$.

<u>A5 :</u>

I=1, j=1, bloc fsi j=1, i=2

I=2, j=1 bloc alors j=1 = j= m+1 ; i=m+3.

J=m+1, i=m+3 :

Si m est pair donc m+3 est impair bloc fsi j=m+1, i=2*m+4

Si m est impair donc m+3 est pair bloc alors j=1 m+1 ; i= 2*m+4

I=2*m+4 bloc alors j=1 j=m+1 ; i=3*m+5

……

I > n.

- Si m > n, on fera les deux premières itérations uniquement de la boucle externes (i=1, i=2). Dans l'itération 2, la nouvelle valeur de i dépasse la condition d'arrêt de la boucle, donc le nombre itération = m+2 (m pour la boucle interne et 2 pour la boucle externe).
- Si n > m, alors le nombre d'itérations au totale <= N.

Au pire cas, à chaque itération de la boucle externe, la valeur de i modulo 2 =0 ; c'est-à-dire que la boucle interne sera exécuter à chaque itérations, on aura donc, I=1, 2, m+3, 2m+4, 3m+5, …. N. chaque m est obtenu en m itérations de la boucle interne.

F(A5) < N   O(n).