TD OFI. Complexité. Année 1997.

exercice 1:

Soient 2 Procédures $P1$ et $P2$ de complexité respective $2n^2$ et $100n$
10^7 instructions sont effectuées par seconde.

- Quelle taille maximum des données peut être traitée par les deux procédures en 1 minute, 10 minutes, 1 heure?
- Pour une taille de données n égal à 10^2, 10^3, 10^5, 10^9 quel est le temps de traitement des deux procédures?

exercice 2:

Quelle est la valeur de la variable `counter` à la sortie des morceaux de programme ci-dessous, avec n=4 puis n=15?

```
n;                  n;
counter:=10;        counter:=10;
for i:=1 to n do    for i:=1 to n do
 for j:=i to n do    for j:=i+1 to n do
  counter:=counter+1  counter:=counter+1
 end for             end for
end for             end for
```

Quelle est la valeur de la variable `counter` à la sortie des morceaux de programme ci-dessous?

```
n:=15;                  n:=15;
counter:=10;            counter:=10;
for i:=1 to n do        for i:=1 to n do
 for j:=i+1 to n do      for j:=i to n do
  for k:=j+1 to n do      for k:=j to n do
   counter:=counter+1      counter:=counter+1
  endfor                  endfor
 endfor                  endfor
endfor                  endfor
```

Quel est l'ordre de complexité des quatre algorithmes en fonction de n?

exercice 3:

Tri par bulles Soient n nombres rangés dans un tableau A[1..n]. Calculer la complexité de l'algorithme suivant, au cas le pire:

```
for i:=1 to n-1 do
  for j:=1 to i do
   if(A[j]<A[j+1] then
    ech:=A[j];
    A[j]:=A[j+1];
    A[j+1]:=ech;
   end if
  end for
end for
```

exercice 4:
Quel est la complexité des algorithmes suivants au cas le pire (`n`,`m` donnés, `odd(n)` retourne `True` si n est impaire, `False` sinon):

Algorithme 1

```
if(odd(n)=True)then
 for i:=1 to m do
  sum1:=sum1+1;
 end for
else
 for i:=1 to n do
  for j:=1 to n do
   sum2:=sum2+1;
  end for
 end for
end if
```

Algorithme 2

```
matrice-au-carre(n,A[n,n])
integer i,j,k;
for i:=1 to n do
  for j:=1 to n do
   C[i,j]:=0;
   for k:=1 to n do
     C[i,j]:=C[i,j]+A[i,k]*A[k,j]
   end for
  end for
end for
```

exercice 1:

t=1 minute	NP1=6.10^6	NP2=$rac(3).10^4$
t=10 minutes	NP1=6.10^7	NP2=$rac(30).10^4$
t=1heure	NP1=$3,6.10^8$	NP2=$rac(1,8).10^5$
N=10^2	TP1=$10^{-3}sec$	TP2=$2.10^{-3}sec$
N=10^3	TP1=$10^{-2}sec$	TP2=$2.10^{-1}sec$
N=10^6	TP1=$10sec$	TP2=55heures 33 minutes 20 secondes
N=10^9	TP1=2heures 46 minutes 40 sec	TP2=15220 siecles 7ans ....

exercice 2:

$$\sum_{i=k}^{n} C_i^k = C_{n+1}^{k+1}$$

Premier Algorithme
valeurCompteur=$\alpha + \sum_{i=1}^{n} C_i^1 = C_{n+1}^2$
$\alpha = 10, n = 15, valeurCompteur = 10 + 15.16/2 = 130$
Second Algorithme
valeurCompteur=$\alpha + \sum_{i=1}^{n-1} C_i^1 = C_n^2$
$\alpha = 10, n = 15, valeurCompteur = 10 + 15.14/2 = 115$
Troisième Algorithme
valeurCompteur=$\alpha + \sum_{i=2}^{n-1} C_i^2 = C_n^3$
$\alpha = 10, n = 15, valeurCompteur = 15.14.13/6 = 465$
Quatrième algorithme
valeurCompteur=$\alpha + \sum_{i=2}^{n-1} C_i^2 + \sum_{i=1}^{n} C_i^1 + \sum_{i=1}^{n-1} C_i^1 = 10 + 120 + 105 + 455 = 690$

Complexité des algorithmes

Algorithme 1	
C1=n(n+1)/2	L'algorithme est en $O(n^2)$
Algorithme 2	
C2=n(n-1)/2	L'algorithme est en $O(n^2)$
Algorithme 3	
C3=(n+1)n(n-1)/2	L'algorithme est en $O(n^3)$
Algorithme 4	
C4=C1+C2+C3	L'algorithme est en $O(n^3)$

exercice 3:
Nombre d'opérations possibles au cas le pire: 4.n(n-1)/2
L'ordre de complexité est donc $O(n^2)$

Pour ceux qui auraient fini en avance
exercice 4:
Algo1: $O(max(m, n^2))$
Algo2: nombre d'opérations: $n^2 + 3.n^3$, $O(n^3)$