Exercices Supplémentaires sur la Complexité des Algorithmes

Exercice 1 : Analyser la complexité de chacun des algorithmes suivants en donnant une brève explication à votre réponse.

Algorithme	Complexité	Explication
Somme de deux vecteurs de n éléments		
Somme de deux matrices carrées (n, n)		
Produit de deux matrices carrée (n, n)		
Calcul itératif du factoriel de n		
Calcul itératif de la somme 1+ 2+ ….+ n		
Insertion en tête d'une liste chainée simple de taille n		
Insertion en tête d'un tableau de taille n		
Insertion à la fin d'une liste chainée simple de taille n		
Insertion à la fin d'un tableau de taille n		
Tester si un nombre est pair ou impair		

Exercice 2 : Analyser la complexité des algorithmes suivants :

`void exp1() {`
`  int i ;`
`  for (int i = 1 ; i <= 100 ; i++)`
`  cout<<i*2 ;`
`}`	

`void exp2(int n, int s) {`
`  int p = n ; s = 0 ;`
`  while (p >= 1) {`
`  int i = 1 ;`
`  while (i <= n) {`
`  s = s + p ; i++ ;`
`  }`
`  p = p /2 ;`
`  }`
`}`	

`void exp3(int n) {`
`  for (int i = 1 ; i <= n ; i++)`
`  for (int j = 1 ; j <= i ; j++)`
`  exp1() ;`
`}`	

`void exp4() {`
`  int i ;`
`for(int i = n ; i > 0 ; i=i/2) {`
`  for(int j = 1 ; j < n ; j=j*2) {`
`  for(int k = 0 ; k < n ; k=k+2) {`
`  //un nombre constant`
`  d'instructions`
`  }`
`  }`
`}`	

`for (i=0; i<=n-1; i++){`
`  for (j=i+1; j<=n-1; j++){`
`  //un nombre constant`
`  d'instructions`
`  }`
`}`	

Exercice 3 : analyser la complexité des fonctions récursives suivantes:

```
int somme (int n) {
 if (n==0) return 0 ;
else return n + somme (n-1)
}
```

1) Formule de récurrence

.....
.....
.....
.....
.....

2) Résolution par la méthode de substitution et calcul de la complexité

.....
.....
.....
.....
.....
.....

```
Algorithme F
Entrée : tableau T de taille n, les indices sont [1..n]
Début
  Si n  == 1 alors afficher T[1]
  Sinon  début
            afficher(T[n]) ;
            F (Tableau T privé du dernier élément) ;
            F (Tableau T privé du premier élément) ;
          Fin
Fin
```

2) Formule de récurrence

.....

.....

.....

.....

.....

.....

.....

.....

.....

2) Résolution par la méthode de substitution et calcul de la complexité

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Exercice 4

Analyser la complexité de F au pire et au meilleur cas :

```
bool F(int T[], int n) {
  for (int i = 0; i < n-1; i++)
    if (T[i] == T[i+1])return true;
  return false;
}
```

Pire cas	Meilleur cas