SÉRIE N°5 : COMPLEXITÉ ALGORITHMIQUE (SUITE)

Exercice 12: Donner la complexité de la fonction récursive suivante :

```
def factoriel(n) :
        if (n == 0) : return 1
        else : return n*factoriel(n-1)
```

Exercice 13: Donner la complexité de la fonction récursive suivante :

```
def RD(L,x,inf,sup) :
        if sup<inf:
                return len(L)
        else:
                m=(inf+sup)//2
                 if L[m] ==x:
                        return m
                if L[m] < x:
                        return RD(L,x,m+1,sup)
                if L[m] > x:
                        return RD(L,x,inf,m-1)
```

Exercice 14: Donner la complexité pour chaque fonction récursive suivante

Formule de récurrence de la fonction	Complexité
C(n)=4C(n/2)+n	
C(n)=2C(n/2)+n	
C(n) = 3C(n/4) + n	

Exercice 15: Donner la complexité de ce Tri :

Tri	Complexité

def fusion(L1,L2) :
if L1==[] :return L2
if L2==[] :return L1
if L1[0]<L2[0] :
return [L1[0]]+fusion(L1[1 :],L2)
else :
return [L2[0]]+fusion(L1,L2[1 :])
def tri_fusion(L):
if len(L) <=1: return L
return fusion(tri_fusion(L[:len(L)//2]),tri_fusion(L[len(L)//2:]))	