

Algorithmique et Structures de données 1

LIV 2022-2023
Travaux Dirigés 5

Site du cours : <https://defelice.up8.site/algo-struct.html>

Les exercices marqués de (@) sont à faire dans un second temps.

Exercice 1. Suite

Voici quatre algorithmes qui résolvent le même problème dont on a calculé la complexité dans le pire des cas.

nom	complexité en temps	complexité en espace
A	$\Theta(n^2)$	$\Theta(n \log_2(n))$
B	$\Theta(n^3)$	$\Theta(n)$
C	$\Theta(2^n)$	$\Theta(\log_2(n))$
D	$\Theta(\log_2(n)n^{0,05})$	$\Theta\left(n^{\frac{1}{10}}\right)$

Répondre aux questions suivantes en se fiant seulement au tableau.

1. Classer selon le nombre d'opération les algorithmes du meilleur au moins bon ?
2. Classer selon la mémoire occupée les algorithmes du meilleur au moins bon ?

Exercice 2. Complexité

Voici trois fonctions :

```
int ostap(int n)
{
    int S=0,i;
    for(i=0;i<n;i++)
        S+=i;
    return S;
}
```

```
int tarass(int n)
{
    int S=0,i;
    for(i=0;i*i<n;i++)
        S+=i;
    return S;
}
```

```
int boulba(int n)
{
    int S=0,i;
    for(i=0;i<n*n;i++)
        S+=i;
    return S;
}
```

Pour chacun des trois algorithmes associés aux trois fonctions précédentes, quelle est sa complexité en temps et en espace, dans le pire des cas et en moyenne (à donner en fonction de n) ?

Exercice 3. Somme liste

Écrire une fonction `somme` faisant la somme des éléments d'une liste chaînée.

Exercice 4. Ajoute en queue

Écrire une fonction `void ajouteQueue(int val, cellule_t** liste)` qui ajoute en queue de liste la valeur `val`. Voici un exemple d'utilisation de la fonction :

```
.
cellule_t* liste=NULL; // vide au départ
```

```
.  
. // construction de la liste  
.ajouteQueue(5,&liste); // ajoute l'élément 5 à la fin de la liste
```