

TD4 - DEV5.1 Qualité algorithmique

Complexité algorithmique

Ex 1 - Création du repo Git

- Création d'un repo par étudiant
Copier : https://grond.iut-fbleau.fr/menault/TD4_DEV51_Qualite_Algo
Nom du repo : TD4_DEV51_<login1>

Ex 2 - Calculs de complexité de fonctions

Calculez la complexité des fonctions suivantes :

```
def function_1(tableau1,tableau2):
    presentDansDeuxListes = 0
    for nombre1 in tableau1:
        for nombre2 in tableau2:
            if nombre1 == nombre2:
                presentDansDeuxListes += 1
                break
    return presentDansDeuxListes
```

```
def function_2(x):
    valeur = 0
    while x > 0:
        valeur = valeur + x
        x -= 1
    return valeur
```

```
def function_3(x):
    valeur = 0
    if x < 0:
        valeur = -x
    if x == 0:
        pass
    if x > 0:
        valeur = x
    return valeur
```

Ex 3 - student_rank.c

- Reprendre le code fourni dans le TD1 (la version sans optimisation)
- Calculer la complexité algorithmique de la fonction "sort_students", on suppose que :
 - La complexité de "malloc" est $O(1)$
 - La complexité de "free" est $O(1)$

Ex 4 - Algorithme de tri

- Créer un algorithme pour trier un tableau à N dimensions de M valeurs
- Chaque dimension doit être trié par ordre croissant de la somme des valeurs de la dimension inférieur
- Exemple pour un tableau à 2 dimensions de 3 valeurs
[[0,3,2], [9,4,5], [4,1,3]] \rightarrow [[0,2,3], [1,3,4], [4,5,9]]
- Calculer la complexité algorithmique de l'algorithme écrit