

# ENSAE TD noté, mardi 3 novembre 2021

*Le programme devra être envoyé par mail au chargé de TD et au professeur.*

## 1

On s'intéresse à l'optimisation de la multiplication de plusieurs matrices.

### Prérequis

On a besoin d'une fonction qui mesure le temps d'exécution d'une fonction.

```
import time

def mesure_temps_fonction(fct, N=100):
    begin = time.perf_counter()
    for i in range(N):
        fct()
    return (time.perf_counter() - begin) / N

mesure_temps_fonction(lambda: time.sleep(0.1), N=10)
```

1) Pourquoi  $(m1 @ m2) @ m3$  est-il plus lent que  $m1 @ (m2 @ m3)$  avec les matrices suivantes? (2 points)

```
import numpy

m1 = numpy.random.randn(997, 93)
m2 = numpy.random.randn(93, 1003)
m3 = numpy.random.randn(1003, 97)

mesure_temps_fonction(lambda: m1 @ m2 @ m3)
```

2) Ecrire une fonction qui calcule le nombre d'opérations dans une multiplication de deux matrices (2 points)

```
def n_ops(m1_shape, m2_shape):
    # ...
```

3) Ecrire une fonction qui retourne le meilleur coût d'une multiplication de trois matrices et la meilleure option. (2 points)

```
def n_ops_3(sh1, sh2, sh3):
    # ...
    return n_operation, (1 ou 2)
```

4) Ecrire une fonction qui effectue le produit de trois matrices le plus rapidement possible (2 points)

```
def produit3(m1, m2, m3):
    #...
    return
```

5) Vérifiez que vous retrouvez les mêmes résultats avec la fonction `mesure_temps` (2 points)

**Ici, vous avez le choix entre faire les questions 6 à 9 ou les questions 9 et 10.**

6) Ecrire une fonction qui retourne le meilleur coût d'une multiplication de 4 matrices et la meilleure option. La fonction doit appeler `n_ops_3`. (3 points)

```
m4 = numpy.random.randn(97, 20)

def n_ops_4(sh1, sh2, sh3, sh4):
    # ...
    return n_operation, (1 ou 2 ou 3)
```

7) Ecrire une fonction qui effectue le produit de 4 matrices le plus rapidement possible. La fonction doit appeler `produit3`. (3 points)

```
def produit4(m1, m2, m3, m4):
    # ...
    return
```

8) Vérifiez que vous retrouvez les mêmes résultats avec la fonction `mesure_temps` et la matrice `m4`. (2 points)

9) On se penche sur le cas à une multiplication de  $N$  matrices, combien y a-t-il de multiplications de 2 matrices ? (2 points)

**Ici s'arrête l'énoncé pour ceux qui ont choisit de répondre aux question 6 à 9. L'objectif de ces premières questions est de préparer le cas d'une multiplication de  $N$  matrices.**

10) Résoudre l'optimisation de multiplication de  $N$  matrices.