**Grand Oral**

Maths: Notation big O ou complexité algorithmique.

En 2009, une entreprise sud-africaine à envoyer à un serveur situé 97 km plus loin un fichier, en même temps elle a lâché vers la même destination un pigeon voyageur équipé d’une clé USB contenant le même fichier. Et bien sûr, sinon ce ne serait pas drôle, le vainqueur de cette course, avec 1 heure d’avance qui plus est, était le pigeon.

Bonjour, je m’appelle Adje Axel et aujourd’hui je vais vous parler d’un outil mathématique qui vous aidera à comprendre pourquoi votre internet est si lent, pourquoi les recherches google sont si rapides, ou encore pourquoi les ordis de l’école prennent 20min à connecter les élèves. Aujourd’hui je vais vous parler de la complexité algorithmique.

plsrs types de complexité:

1. Complexité constante: O(1)
2. complexité linéaire: 0(n)
3. complexité quadratique: 0(n^2)
4. complexité logarithmique: O(log(n))

Exemple de différentes complexités:

Linéaire:

def getSum**(**n**)**:

sum = 0

**for** number **in** range**(**1, n + 1**)**:

sum += number

**return** sum

Constante:

**def** getSum**(**n**)**:

**return** n \* **(**n + 1**)** / 2

Quadratique:

**def** exempleOn2**(**n**)**:

**for** number **in** range**(**n**)**:

**for** number **in** range**(**n**)**:

print**(**number + 1**)**

Logarithmique:

**def** exempleOlogn**(**n**)**:

i = 1

**while(**i **<** n**)**:

i = i \* 2

print**(**i**)**

Conclusion: les maths jouent encore aujourd’hui un rôle essentiel en informatique, la notation Big O n’est qu’un exemple parmi beaucoup d'autres. Sans elle, il serait dur de se rendre compte de l'efficacité et de la scalabilité de certains codes, et c’est avec ce genre d’information qu’ il n’y aurait pas de facebook, pas d’instagram, pas de google… On ne se rend pas compte de la quantité d'optimisation derrière la technologie que l’on utilise.