Python ITÉRATION 1

Introduction à l'algorithmique

1.1 — Algo avec des cartes

1h à 2h — Par îlot

Un jeu de cartes et un problème vous seront donnés. Travaillez en équipe, par îlot, afin de proposer une conception préliminaire et détaillée (c.à.d. écrivez les signatures, et le pseudocode associé à votre solution) suite à l'analyse du problème.

⚠ *Le langage pseudocode n'a pas été normalisé, à vous de choisir la norme qui rend votre algorithme le plus lisible, compréhensible pour vous et auprès des autres.*

⚠ *Il est conseillé de réaliser votre conception détaillée avec une numération basée sur l'index zéro. Cela vous sera utile lors du passage au codage, car Python est un langage qui utilise cette convention.*

RESSOURCES

- Pourquoi indexer des éléments d'un tableau à partir de zéro ? Voici une lettre écrite par le mathématicien et informaticien Dijkstra en 1982. Dijkstra est notamment reconnu pour ses travaux sur les graphes. Un algorithme de recherche du plus court chemin porte son nom.
  Ressource historique : https://www.cs.utexas.edu/users/EWD/ewd08xx/EWD831.PDF

COMPÉTENCES ASSOCIÉES

- Méthodologie de conception d'une solution informatique

Livrables

➔ Le pseudocode répondant au problème posé (à mettre en commentaire de la compétence associée sur Campus Skills)

1.2 — En Python, ça donne quoi ?

2h — Individuel

Après validation par le formateur de votre conception, réalisez le codage de votre algorithme avec le langage Python.

1.3 — Les tris classiques

5h — Individuel

Récupérez le jupyter notebook `AlgosTriClassiques.ipynb`. Vous allez maintenant programmer quelques algorithmes de tri classiques :

- Bubble Sort
- Insertion sort
- Selection sort

Pour chaque algorithme, êtes-vous capable de :

- Déterminer si votre méthode est une fonction ou une procédure ?
- Calculer la complexité O de chaque algorithme ?
- Analyser l'exécution de vos algorithmes à l'aide des « magic commands ». `%prun` ou `%lprun` pourrait être utile pour identifier la complexité.

RESSOURCES

- Visualisation des algorithmes de tri
  https://www.cs.usfca.edu/~galles/visualization/ComparisonSort.html
- La notation grand O
  https://marcarea.com/weblog/2019/01/21/complexite-algorithmique-et-notation-grand-o
- Les « magic commands »
  https://jakevdp.github.io/PythonDataScienceHandbook/01.07-timing-and-profiling.html

COMPÉTENCES ASSOCIÉES

- Analyse de la complexité d'un algorithme
