

Big Data. TD 2












Sergey Kirgizov

ESIREM

Complexité des algorithmes

L'objectif est de déterminer empiriquement les capacités maximales d'un système de calcul. Pendant ce TD vous découvrez ce que la Big Data signifie dans votre cas.

Veuillez utiliser votre langage préféré.

-  EXERCICE 1 : Réaliser un algorithme de tri de votre choix : tri à bulles, tri fusion, tri par insertion ou autre. Le programme doit accepter à l'entrée un fichier et créer un autre fichier avec des lignes triées du premier fichier.
-  EXERCICE 2 : Préparer les fichiers de tailles différentes : 1 000 lignes, 5 000 lignes, 100 000 lignes, 500 000 lignes, 1M lignes, 1.5M lignes, 2M lignes, ..., 10M lignes. Vous pouvez utiliser par exemple ce fichier <https://kirgizov.link/teaching/esirem/bigdata-2021/dataset/wikirank-fr.tsv.gz>.
-  EXERCICE 3 : Lancer votre programme en lui donnant les fichiers des différentes tailles tout en mesurant le temps du calcul nécessaire.
-  EXERCICE 4 : Visualiser la dépendance entre la taille du fichier et le temps du tri.
-  EXERCICE 5 : Trouver une approximation de la fonction de la dépendance.
-  EXERCICE 6 : Utiliser l'approximation pour prédire le temps nécessaire pour trier un fichier contenant 20M ou 30M lignes.
-  EXERCICE 7 : Vérifier vos prédictions.
-  EXERCICE 8 : Faites une prédiction de la taille de fichier maximale que vous pouvez traiter sur votre ordinateur en une journée.
-  EXERCICE 9 : Que se passe-t-il si nous mesurons la taille du fichier non pas en lignes mais en octets ?
-  EXERCICE 10 : Mettre en œuvre la même analyse pour la quantité maximale de mémoire utilisée.
-  EXERCICE 11 : Mettre en œuvre la même analyse pour l'algorithme "Tri par fusion parallèle" https://en.wikipedia.org/wiki/Merge_sort#Parallel_merge_sort