

TP noté : Algorithme de tri hybride

Le tri par insertion est plutôt meilleur que le tri rapide sur des petites listes ou sur des listes presque triées. En conséquence, pour trier une liste, on peut utiliser le tri rapide puis, en dessous d'un seuil n_0 à déterminer, on utilise le tri par insertion. Ce TP a pour objectif de déterminer ce seuil n_0 .

Vous devez utiliser le module `mesFonctionsTri.py`.

Votre travail de TP doit être stocké dans un fichier nommé par votre prénom et nom sous le format suivant : `triHybridePrenomNom.py`

1. Écrire une fonction de tri prenant en paramètres une liste L et un seuil n_0 . Cette fonction trie L en utilisant le tri rapide puis en basculant au tri par insertion pour le tri des sous-listes de taille inférieure ou égale à n_0 .
2. Générer aléatoirement une liste l de taille 50000 dont les termes sont des réels choisis aléatoirement dans l'intervalle $[-1000, 1000]$. Représenter graphiquement l'évolution du temps d'exécution de ce tri hybride sur la liste l en fonction du seuil n_0 , avec n_0 variant entre 1 et 50, avec un pas de 5. Puis, afficher la valeur optimale de n_0 , c'est-à-dire la valeur de n_0 induisant le plus petit temps de réponse. Afficher la différence entre le plus petit temps de réponse et le temps d'exécution du tri rapide seul (sans hybridation).
3. Une autre façon de procéder est d'utiliser le fait que le tri par insertion soit rapide sur des listes presque triées. Une variante du basculement précédent consiste donc à stopper le tri rapide sur des sous-listes de taille inférieure à n_0 , en laissant les sous-listes telles quelles dans un premier temps. Ainsi, ce tri partiel retourne une liste presque triée dans le sens où tous les termes seront à une distance au plus n_0 de leur position finale (la liste est triée par tranches d'au plus n_0). On lance alors un tri par insertion sur la liste obtenue.
Ecrire une deuxième fonction de tri hybride qui prend en paramètres une liste L et un seuil n_0 et trie L selon ce principe. Représenter graphiquement (on utilisera la même figure) l'évolution du temps d'exécution de ce nouveau tri hybride de la même liste l que la question 2 en fonction du seuil n_0 , avec n_0 variant entre 1 et 50, avec un pas de 5. Sauvegarder ce graphique dans un fichier au format png nommé par votre prénom, nom sous le format : `prenomNom1.png`. Puis, afficher la valeur optimale de n_0 , c'est-à-dire la valeur de n_0 induisant le plus petit temps de réponse. Afficher la différence entre le plus petit temps de réponse et le temps d'exécution du tri rapide seul (sans hybridation).
4. Pour que votre étude expérimentale soit pertinente, il faudrait comparer vos temps de calcul et construire vos courbes, non pas sur la base d'une unique instance de taille 50000 mais sur la base de plusieurs instance de taille 50000 (au moins 10). Compléter votre programme pour permettre une telle analyse sur la base des temps de calcul moyen. Représenter graphiquement les temps d'exécution moyens de ces deux algorithmes de tri hybrides en fonction du seuil n_0 , avec n_0 variant entre 1 et 30, avec un pas de 3. Sauvegarder le graphique présentant vos résultats dans un fichier au format png nommé par votre prénom, nom sous le format : `prenomNom10.png`.