

DS 2

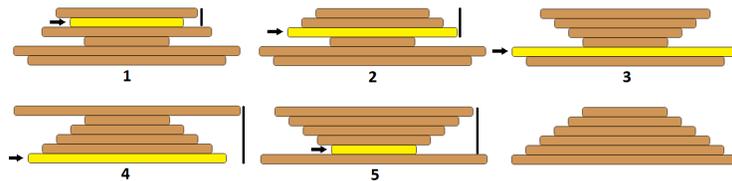
Informatique pour tous, deuxième année

Julien REICHERT

Exercice 1 (question de cours) : Écrire un des algorithmes de tri parmi le tri fusion et le tri rapide, au choix dans sa version en place ou non en place. Prouver sa terminaison ; calculer sa complexité dans le pire des cas par une analyse rigoureuse. Prouver aussi la correction de l'algorithme. En plus de cela, l'algorithme doit avoir un argument booléen précisant si le tri doit se faire dans l'ordre croissant ou non. La fonction `reversed` et la méthode `reverse` sont interdites, de même que la méthode `sort` et toutes les autres astuces du genre, bien évidemment.

Exercice 2 : Tri des crêpes : Sans se pencher sur le problème mathématique (ouvert) qui cherche à déterminer le plus grand (sur l'ensemble des permutations possibles) nombre minimal d'opérations effectuées par un algorithme associé, on souhaite trier une liste en ne s'autorisant qu'une opération sur les listes : renverser une tranche de la liste allant d'une certaine position à la fin.

Écrire un algorithme sous cette contrainte et justifier que la complexité est un $\mathcal{O}(n)$ en nombre de « renversements » (chacun étant de complexité linéaire au pire) pour une liste de taille n .



Exercice 3 : On considère l'algorithme suivant : pour des zones de taille deux, on échange les éléments au besoin, pour des zones de taille inférieure, on ne fait rien, et sinon on trie les deux tiers de gauche, puis on trie les deux tiers de droite, puis on trie de nouveau les deux tiers de gauche (à chaque fois on arrondit par excès pour la sécurité), le tout se faisant toujours en place¹.

Écrire cet algorithme en Python, prouver qu'il termine et qu'il s'agit bien d'un tri puis déterminer sa complexité à l'aide d'une formule de récurrence.²

Exercice 4 : Écrire une version du tri par insertion procédant par dichotomie pour trouver l'endroit où placer chaque élément successif de la liste à trier. L'algorithme peut agir au choix en place ou sur de la mémoire extérieure. Déterminer la complexité asymptotique du programme écrit, d'abord en nombre d'affectations, puis en nombre de comparaisons. Il suffit de donner la formule de récurrence et d'utiliser le cours du chapitre précédent.

Exercice 5 : Écrire une fonction implémentant le « tri Tom », qui consiste à créer à partir d'une liste quelconque de taille notée n la liste des nombres de 0 à $n-1$ dans le même ordre que les éléments de la liste. En cas de doublon l'ordre est au choix, mais pour simplifier il est autorisé de supposer qu'il n'y a pas du tout de doublon. Il est bien entendu interdit de modifier la liste en argument.

Par exemple, l'appel `tritom([6, 3, 5, 2])` retournera la liste `[3, 1, 2, 0]`.

Toutes les méthodes et astuces imaginables sont exceptionnellement autorisées.

1. pas de pénalité si ce n'est pas le cas, mais il faut savoir que c'est plus difficile à écrire
2. La résolution de la récurrence n'est pas exigée pour les étudiants en option SI.

Dans le sixième exercice, nous allons étudier une base de données. La structure associée est la suivante :

- Table Etudiants, avec les attributs Id (clé primaire avec auto-incrémentation, entier naturel), Classe (chaîne de caractères), Nom (idem) et Prenom (idem). On peut considérer que (Nom, Prénom) est également une clé.
- Table Examens, avec les attributs Id (clé primaire avec auto-incrémentation, entier naturel), Date (chaîne de caractères au format `aaaa/mm/jj`) et Matiere (chaîne de caractères).
- Table Notes, avec les attributs Etudiant (entier naturel), Examen (entier naturel) et Note (entier naturel). Le couple (Etudiant, Examen) est une clé.

Exercice 6 : Écrire une requête SQL permettant de déterminer...

- ... le nombre d'examens d'informatique [question 6a].
- ... la date du premier examen [question 6b].
- ... le nombre de matières où un examen a été enregistré [question 6c].
- ... le nombre de notes au-dessus de dix sur l'ensemble des examens passés en MPSI 2 [question 6d].
- ... la moyenne en informatique de Théo Dupont [question 6e].
- ... le nom de l'étudiant ayant la meilleure moyenne brute (on ne tient pas compte des matières, donc) [question 6f].