Diviser pour Régner

Objectif : illustrer le paradigme "Diviser et Régner" en revisitant un algorithme "classique".

Comparer des listes de préférences

A un utilisateur est associée une liste complétement ordonnée de préférences pour n "articles". Pour pouvoir définir les profils d'utilisateurs, par exemple pour une recommandation efficace, on cherche à associer à chaque utilisateur le profil d'utilisateur type le plus proche et donc à mesurer la distance entre deux listes des préférences. Pour simplifier, **les listes de préférences considérées seront des permutations de 0 à n-1**.[1]

Q 1. Quelle distance ?

Soient les trois profils :
A : 0 3 5 1 4 2
B : 2 4 3 1 5 0
C : 0 5 1 3 4 2

Entre B et C, quel profil vous paraît le plus proche de A ? Pourquoi ? Que proposeriez-vous pour mesurer la distance entre deux listes de préférences ?

Q 2. Une inversion de préférences entre X et Y est un couple $(i, j)(i \neq j)$ tel que X préfère i à j et Y préfère j à i. On définit la distance entre deux listes de préférences comme étant le nombre d'inversions de préférences entre X et Y. Par exemple, pour A et C il y a deux inversions :
(3,5) A préfère 3 à 5, B préfère 5 à 3.
(3,1) A préfère 3 à 1, B préfère 1 à 3.

Quelles sont les inversions entre A et B ?
Combien d'inversions y a-t-il au maximum entre deux listes de préférences pour n articles ?

Q 3. Pour simplifier le calcul du nombre d'inversions entre des listes et une liste fixée, on normalise par rapport à cette liste fixée, en renumérotant les articles selon celle-ci. Par exemple, si on normalise par rapport à A,- l'article 0 garde le numéro 0, l'article 3 devient le numéro 1, le 5 devient le 2, ...- on obtient :
pour B : 5 4 1 3 2 0, pour C : 0 2 3 1 4 5
Formellement, la $i$ème valeur de $Y = Y_0, \ldots, Y_{n-1}$ normalisée par rapport à $X = X_0 \ldots, X_{n-1}$ est l'unique j tel que $Y_i = X_j$.
*R*appel : Une **inversion** dans une liste de n éléments, $l_0, \ldots, l_{n-1}$ est un couple (i, j) tel que $i < j$ et $l_i > l_j$.
Soient X et Y deux listes de préférences et Y_X la normalisée de Y par rapport à X.
Quel est le lien entre le nombre d'inversions de Y_X et la distance entre X et Y ?

Algorithme et Implémentation

L'objet central du TP est donc des listes de préférences qui seront des permutations des entiers de 0 à n-1. On choisira de les implémenter sous forme de tableau. Vous pouvez choisir d'implémenter en Java, C ou Caml.

Q 4. *Normalisation* Proposer un algorithme linéaire qui normalise une liste par rapport à une autre.
Q 5. *Calcul du nombre d'inversions*
Q 5.1. Proposer et implémenter un algorithme qui calcule le nombre d'inversions d'une liste en $\Theta(n^2)$
Q 5.2. En utilisant la paradigme "diviser pour régner", proposer un algorithme en $\Theta(n \log n)$.
Aide : S'inspirer du tri fusion.
Q 6. A partir des questions précédentes, implémentez deux méthodes pour le calcul de la distance entre deux listes de préférences.
Q 7. Expérimenter les deux méthodes. Comparer les temps d'exécution et interpréter.
A rendre : Une archive avec le code commenté, un fichier texte avec les réponses aux questions 2 et 3 et les conclusions sur l'expérimentation. Des fichiers de données sont disponibles pour le test. Pour simplifier le test, respectez leur syntaxe. Par exemple, si vous avez choisi Java, on pourra tester par `java` *votreTP* < *donnees*.

1. Une permutation d'un ensmeble X est une bijection de X sur X. Une permutation σ de 0 à n-1 peut donc être donnée par $\sigma(0), \ldots, \sigma(n-1)$, i.e. une liste de n entiers telle que chaque entier de 0 à $n-1$ soit présent une et seule fois.