

Travaux pratiques #3 : QuickSort et MinCut

L'objectif de ce TP est d'observer le comportement des algorithmes `MinCut` et `Quicksort`.

Les questions d'expérimentation sont volontairement très libres : c'est à vous d'imaginer des expériences appropriées, et surtout de justifier vos choix dans le rapport.

MinCut

1. Implanter les algorithmes `RandMinCut` et `BetterMinCut` du cours. Il est conseillé de représenter les graphes par listes d'adjacence.
2. Effectuer une comparaison expérimentale des deux algorithmes : mettre en évidence la différence des probabilités de succès.
Note. Il n'est pas nécessaire d'implanter un *vrai* algorithme de coupe minimale pour cela. On considèrera simplement que la taille de la plus petite coupe renvoyée par `BetterMinCut` est la taille d'une coupe minimale.

Quicksort

1. Implanter l'algorithme `Quicksort`, de préférence « en place ».
2. Modifier l'algorithme pour qu'il renvoie également le nombre de comparaisons effectuées.
3. Mettre en évidence expérimentalement le comportement de l'algorithme (nombre moyen de comparaisons, probabilité d'effectuer sensiblement plus de comparaisons que la moyenne, etc.)
4. Comparer le comportement de l'algorithme sur trois types d'entrée : tableau trié, tableau trié à l'envers, permutation aléatoire. Conclusion ?