
TP4 : Diviser pour régner

Le but de ce TP est d'implanter des algorithmes de type « diviser pour régner » vus en cours. L'objectif est de faire **au minimum les deux premiers exercices**. Faire le troisième est cependant un excellent entraînement !

Exercice 1.*Tri fusion*

Le but de cet exercice est d'implanter le tri fusion, comme étudié en cours.

1. Compléter la fonction `void fusion(int n1, int n2, int* T1, int* T2, int* T)` qui fusionne les tableaux T1 et T2 de tailles respectives n1 et n2 dans le tableau T. On suppose que T a été correctement alloué à la bonne taille précédemment.
2. Compléter la fonction `void trifusion(int n, int* T)`. Il faudra déclarer et allouer deux tableaux T1 et T2 qui seront à la fin fusionnés dans T. Bien penser à supprimer les tableaux T1 et T2 en fin de fonction.
3. Tester votre fonction grâce au fichier `tests.cc`.

Exercice 2.*Calcul de rang*

Le but de cet exercice est d'implémenter un calcul de rang linéaire, comme étudié en cours.

On propose deux choix possibles pour `choixPivot` : un pivot fixe (`T[0]`) et un pivot aléatoire. La fonction `int choixPivot(int n, int* T, pivot P)` implante ces choix et renvoie un pivot de T.

1. Compléter la fonction `int rang(int k, int n, int* T, pivot P)` permettant de trouver le $k^{\text{ème}}$ plus petit élément du tableau T de taille n.
2. Tester la fonction avec le fichier `tests.cc` et comparer les temps de calcul en fonction du choix de pivot : voit-on une différence sur un tableau aléatoire ? sur un tableau initialement trié (ordre croissant ou décroissant) ?

Exercice 3.*Tri rapide (bonus)*

L'objectif de cet exercice est d'implanter l'algorithme de tri rapide. Il se base sur la même idée que le calcul de rang : (i) on choisit un pivot p ; (ii) on construit les deux tableaux T_{inf} et T_{sup} à l'aide de p ; (iii) on trie par appels récursifs ces deux tableaux ; (iv) on reconstruit T trié en concaténant trois tableaux : T_{inf} trié, un tableau de n_{eq} fois le pivot p et T_{sup} trié.

1. Compléter la fonction `void trirapide(int n, int* T, pivot P)` qui implante cet algorithme.
2. Tester la fonction avec le fichier `tests.cc` et comparer les temps de calculs : comparer entre les deux choix possibles de pivot (dans le cas tableau aléatoire, tableau croissant, etc.) et avec le tri fusion.