

Algo I - TD7-8

Tri et Complexité

Frédéric RAYAR

Novembre 2012

1 Des tris, des tris, et encore des tris ...

On considère le tableau \mathbf{T} suivant :

3	5	2	0	1	4
---	---	---	---	---	---

Pour chacun des méthodes de tris ci-dessous :

1. Rappeler le fonctionnement du tri.
2. Dérouler l'algorithme de tri sur \mathbf{T} , en indiquant les étapes intermédiaires.
3. Ecrire en pseudo-code l'algorithme de tri associé.
4. Donner la complexité de cet algorithme.

1.1 Tri à bulles

Aussi appelé *tri par propagation*, **Bubble sort** en anglais.

1.2 Tri par sélection

Aussi appelé *tri par extraction*, **Selection sort** en anglais.

1.3 Tri par insertion

Insertion sort en anglais.

1.4 Tri rapide

Quicksort en anglais.

1.5 Tri fusion

Mergesort en anglais.

2 Complexité

2.1 Fonction puissance

1. Spécifier la fonction *puissance* qui calcule x^n .
2. Calculer sa complexité **exacte**.

2.2 Fonction recherche simple

1. Spécifier la fonction *recherche_simple* qui recherche linéairement un élément dans un tableau (considérer un tableau d'entiers).
2. Calculer sa complexité **dans le meilleur des cas**.
3. Calculer sa complexité **dans le pire des cas**.

2.3 Fonction élément commun

1. Spécifier la fonction *element_commun* qui calcule si il existe au moins un élément commun à deux tableaux (considérer des tableaux d'entiers).
2. Calculer sa complexité **dans le pire des cas**.

2.4 Fonction recherche dichotomique

1. Spécifier la fonction *recherche_dichotomique* (considérer des tableaux d'entiers).
2. Calculer sa complexité **dans le pire des cas**.

2.5 Des tris, des tris, toujours des tris ...

Calculer la complexité de chacun des algorithmes de tris présentés précédemment.