

# Algorithmique TD4:

## Tris simples

30 octobre 2009

1. Tri par insertion :
  - (a) Réécrire le tri par insertion vu en cours pour trier dans l'ordre décroissant.
  - (b) Quelles sont les meilleurs et pires configurations en terme de coût ?
  - (c) Quels sont ces coûts ?
  - (d) Est-il possible d'améliorer le coût de cet algorithme en appliquant une recherche dichotomique (cf TD2) pour l'insertion d'une valeur dans le sous tableau déjà trié.
2. Tri par sélection :

On considère le tri suivant de  $n$  nombres rangés dans un tableau  $A$  : on commence par trouver le plus petit élément de  $A$  et on le permute avec  $A(1)$ . On trouve ensuite le deuxième plus petit élément de  $A$  et on le permute avec  $A(2)$ . On continue de cette manière pour les  $n - 1$  premiers éléments de  $A$ .

  - (a) Écrire l'algorithme de ce tri connu sous le nom de tri par sélection.
  - (b) Quel est l'invariant de boucle de cet algorithme ?
  - (c) Pourquoi suffit-il d'exécuter l'algorithme pour les  $n - 1$  premiers éléments ?
  - (d) Donner les temps d'exécutions du cas moyen et du cas le plus défavorable, exprimés avec la notation  $\Theta$ .

3. Tri à bulles :

Le tri à bulles est un algorithme de tri très populaire. Il consiste à effectuer des permutations répétées d'éléments contigus qui ne sont pas dans le bon ordre, jusqu'à ce qu'il n'y en ai plus.

- (a) Ecrire l'algorithme du tri à bulle en se basant sur la trace d'exécution suivante :

<b>5</b>	<b>2</b>	4	6	1	<b>3</b>
2	<b>5</b>	<b>4</b>	6	1	3
2	4	5	<b>6</b>	<b>1</b>	3
2	4	5	1	<b>6</b>	<b>3</b>
2	4	5	1	3	6
2	4	<b>5</b>	<b>1</b>	3	6
2	4	1	<b>5</b>	<b>3</b>	6
2	4	1	3	5	6
2	<b>4</b>	<b>1</b>	3	5	6
2	1	<b>4</b>	<b>3</b>	5	6
2	1	3	4	5	6
<b>2</b>	<b>1</b>	3	4	5	6
1	2	3	4	5	6

- (b) Modifier l'algorithme précédent pour qu'il s'arrête dès qu'un passage n'a détecté aucune permutation.