

Réponse au premier numéro :

Insertion :

12	4	7	3	6	8	2	10	9
4	12	7	3	6	8	2	10	9
4	7	12	3	6	8	2	10	9
4	7	3	12	6	8	2	10	9
4	3	7	12	6	8	2	10	9
3	4	7	12	6	8	2	10	9
3	2	7	6	12	8	2	10	9
3	2	6	7	12	8	2	10	9
3	2	6	7	8	12	2	10	9
3	2	6	7	8	2	12	10	9
3	2	6	7	2	8	12	10	9
3	2	6	2	7	8	12	10	9
3	2	2	6	7	8	12	10	9
3	2	4	6	7	8	12	10	9
2	3	4	6	7	8	12	10	9
2	3	4	6	7	8	10	12	9
2	3	4	6	7	8	10	9	12
2	3	4	6	7	8	9	10	12

Sélection

12	4	7	3	6	8	2	10	9
2	4	7	3	6	8	12	10	9
2	3	7	4	6	8	12	10	9
2	3	4	7	6	8	12	10	9
2	3	4	6	7	8	12	10	9
2	3	4	6	7	8	9	10	12

Bulle

<b>12</b>	<b>4</b>	<b>7</b>	<b>3</b>	<b>6</b>	<b>8</b>	<b>2</b>	<b>10</b>	<b>9</b>
4	12	7	3	6	8	2	10	9
4	7	12	3	6	8	2	10	9
4	7	3	12	6	8	2	10	9
4	7	3	6	12	8	2	10	9
4	7	3	6	8	12	2	10	9
4	7	3	6	8	2	12	10	9
4	7	3	6	8	2	10	12	9
4	7	3	6	8	2	10	9	12
4	3	6	7	2	8	10	9	12
4	3	6	7	2	8	9	10	12
4	3	6	7	2	8	9	10	12
3	4	6	7	2	8	9	10	12
3	4	6	2	7	8	9	10	12
3	4	2	6	7	8	9	10	12
3	2	4	6	7	8	9	10	12
2	3	4	6	7	8	9	10	12

Temps d'exécution obtenue sur mon ordinateur : i7-4770, 3.40GHz.

Résultat pour le tri plus rapide utilisant le tri insertion :

	0	5	10	15	20	25
10000	3.311	2.249	0.840	0.796	0.793	0.791
100000	47.676	25.463	10.973	10.812	10.889	10.796
1000000	190.904	204.643	175.473	171.881	176.626	172.353
10000000	3304.251	3312.974	3203.449	3244.424	3069.419	3105.602

Résultat pour le tri plus rapide utilisant le tri sélection :

	0	5	10	15	20	25
10000	3.096	2.546	0.896	0.894	0.935	0.926
100000	48.472	27.120	11.794	11.761	11.902	12.013
1000000	199.325	218.526	185.635	179.888	184.382	204.736
10000000	3338.725	3346.567	3289.644	3343.561	3158.441	3253.409

Nous remarquons que l'ajout de cette optimisation donne de très bons résultats, le temps d'exécution sur des petits tableaux (10000 éléments) est divisé par un facteur de 3 à 4. Par contre, pour les tableaux très grands, c'est une différence de moins de 1 milliseconde. Une valeur BARRIERE entre 15 et 25 pour le tri insertion donne des bons résultats, alors que pour le tri sélection une valeur entre 15 et 20 donne des bons résultats. L'utilisation du tri insertion a donné des meilleurs résultats que l'utilisation du tri sélection.

Résultat pour la méthode sort de la classe Arrays :

10000	5.543
100000	49.997
1000000	326.553
10000000	3558.246

Le tri de Java est plus lent que notre version, pourquoi? Le tri de la librairie Arrays est un tri « stable », le nôtre ne l'est pas. Un tri stable est un tri qui assure que deux éléments identiques vont conserver leur ordre. Par exemple, si un tableau contient deux valeurs '7', alors le premier '7' rencontré dans le tableau va rester avant le deuxième '7' lorsque le tri est terminé. Cette caractéristique est importante lorsque nous appliquons plusieurs tris sur différents champs. Par exemple, si nous trions un tableau sur les Prénoms et ensuite sur les Noms, alors nous ne perdons pas l'ordre entre les prénoms lors du tri sur les noms, ce qui n'est pas garanti lorsqu'un tri n'est pas stable.