

Mesures et comparaisons

Complexités des tris

	comparaisons			transferts		
	min	moy	max	min	moy	max
sélection ordinaire	n^2			n		
bulle	n^2			0	n^2	
insertion séquentielle	n	n^2		n	n^2	
insertion dichotomique	n	$n \log n$		0	n^2	
tas	$n \log n$				$n \log n$	
rapide	$n \log n$		n^2	n	$n \log n$	n^2

Mesures effectives

	500 éléments		2000 éléments		8000 éléments	
	qcq.	triés	qcq.	triés	qcq.	triés
sélection ordinaire	0.34 s	0.33 s	5.4 s	5.5 s	1.8 m	1.8 m
bulle	1.5 s	0.36 s	26 s	6.2 s	10 m	2.2 m
insertion séquentielle	0.73 s	0.0 s	13 s	0.06 s	4.8 m	0.22 s
insertion dichotomique	0.52 s	0.0 s	9.2 s	0.01 s	3.9 m	0.04 s
tas	0.05 s	0.07 s	0.35 s	0.38 s	2.0 s	1.9 s
rapide	0.03 s	0.02 s	0.24 s	0.07 s	1.2 s	0.5 s
$n^2/500^2$	1		16		256	
$n \log n / 500 \log 500$	1		5		24	

Les paramètres influents (1)

- influence de la longueur de la liste
 - longueur faible => tous les algorithmes sont acceptables
 - 2 secondes à 10 minutes pour la liste de 8000
 - extrapolation: 5 minutes à 2 mois pour une liste de 800 000
- influence de l'ordre des éléments
 - sélection ordinaire et tas : peu sensibles
 - amélioration d'un facteur constant pour bulle (4) ou rapide (2)
(suppression des transferts)
 - changement d'ordre de grandeur pour les tris par insertion

Les paramètres influents (2)

- influence temps de comparaison et temps de transfert

sélection ordinaire: $n^2/4 (2 T_c) = 1.8 \text{ m}$

insertion séquentielle: $n^2/4 (T_c + T_t) = 4.8 \text{ m}$

insertion dichotomique: $n^2/4 T_t = 3.9 \text{ m}$

$$T_c = 3.3 \mu s$$

$$T_t = 14.5 \mu s$$

tas : $n \log_2 n (2T_c + T_t) = 2 \text{ s}$

rapide : $n \log_2 n (T_c + 0.75T_t) = 1.2 \text{ s}$

Les paramètres influents (3)

- influence du compilateur

sans optimisation par le compilateur => 1.5 à 2.5 fois plus lent

exemple, boucles internes tri séquentiel ordinaire et bulle

```
1.8 m  for L in K + 1 .. La_Liste.Longueur loop
        if La_Liste.Ieme (L) < La_Liste.Ieme (J) then J := L; end if;
        end loop;
2.2 m  for L in reverse K+1 .. La_Liste.Longueur loop
        if La_Liste.Ieme (L) < La_Liste.Ieme (L-1) then Echanger(L - 1, L); end if;
        end loop;
```

sur listes triées, comparaison toujours fausse

- programme de tri standard

temps équivalent au tri rapide

Remarques

- tris par sélection (ordinaire, bulle, tas)
 - les éléments extraits sont à leur place définitive
 - => commencer le traitement sur le résultat avant de tout avoir
 - => tris avec remplacement: sortir l'élément extrait et en réintroduire un autre
- tris par insertion
 - il n'est pas nécessaire de tout avoir pour commencer le tri
 - => faire le tri au fur et à mesure de l'obtention des éléments