

Structure de données et algorithmes

Projet 1: Algorithmes de tri

Pierre GEURTS – Jean-Michel BEGON

27 février 2015

L'objectif du projet est d'implémenter, de comparer et d'analyser les algorithmes de tri suivants:

- (a) L'algorithme INSERTIONSORT (fourni).
- (b) L'algorithme QUICKSORT standard (à implémenter).
- (c) L'algorithme QUICKSORT avec un pivot aléatoire (à implémenter).
- (d) L'algorithme HEAPSORT (à implémenter).
- (e) L'algorithme HYBRIDSORT (décrit ci-après, à implémenter).

1 HybridSort

Afin de combiner les avantages de HEAPSORT (garantie pour le pire cas) et de QUICKSORT (plus rapide en moyenne), on propose l'algorithme HYBRIDSORT reposant sur un compteur d'appels. L'algorithme exécute récursivement le QUICKSORT jusqu'à ce que la profondeur des appels récursifs excède une valeur seuil $K \log_2 N$, où N est la taille (globale) du tableau à trier et K est une constante, dont la valeur par défaut est généralement fixée à 2. L'algorithme passe alors à un HEAPSORT. En outre, lorsque le nombre d'éléments à trier passe en dessous de 15 lors de la phase de QUICKSORT (quelle que soit la profondeur), le tri est délégué à INSERTIONSORT.

HYBRIDSORT repose donc sur QUICKSORT, HEAPSORT (lorsque la récursion est trop longue) et INSERTIONSORT (lorsque les tableaux sont petits).

2 Analyse théorique

Dans votre rapport, nous vous demandons de répondre aux questions suivantes:

- 1.a. Donnez le pseudo-code de HYBRIDSORT (en supposant les autres algorithmes disponibles).
- 1.b. Etudiez sa complexité en temps dans le meilleur et dans le pire cas en fonction de la taille N du tableau à trier. Expliquez à quoi correspondent les meilleur et pire cas et justifiez la complexité que vous obtenez dans ces deux cas.
- 1.c. Est-ce que cet algorithme de tri est stable ? Justifiez brièvement votre réponse.

3 Analyse expérimentale

Dans votre rapport, nous vous demandons également de répondre aux questions suivantes:

- 2.a Reportez graphiquement l'impact de la constante K du HYBRIDSORT sur le temps de tri moyen pour des tableaux (i) aléatoires, (ii) presque triés et (iii) comportant seulement quelques éléments différents.
- 2.b Commentez ces résultats. Sur base de ces données quelle valeur de K recommanderiez-vous lorsqu'on ne peut pas faire d'hypothèses sur les données?
- 2.c Reportez dans le tableau les temps de tri moyens des différents algorithmes:

Tableau	aléatoire	presque trié	peu d'elem. uniques
INSERTIONSORT			
QUICKSORT (standard)			
QUICKSORT (pivot aléatoire)			
HEAPSORT			
HYBRIDSORT ($K = 2$)			

- 2.d Commentez ces résultats.

Remarques:

- Vous devez choisir une taille de tableau pertinente pour l'analyse.
- Les fonctions pour générer les tableaux vous sont fournies dans le fichier `Array.c`.
- Les temps reportés doivent être des temps moyens établis sur base de 10 expériences.

4 Deadline et soumission

Le projet est à réaliser **individuellement** pour le **lundi 16 mars 2015 à 23h59** au plus tard. Le projet est à remettre via la plateforme *cicada* : <http://cicada.run.montefiore.ulg.ac.be/>. Il doit être rendu sous la forme d'une archive `tar.gz` contenant:

- (a) Votre rapport (5 pages maximum) au format PDF. Soyez bref mais précis et respectez bien la numérotation des (sous-)questions.
- (b) Le fichier `Sort.c` complété.

Les fichiers suivants vous sont fournis:

- `Array.h` et `Array.c`: une petite bibliothèque pour générer différents types des tableaux.
- `Sort.h`: le header contenant le prototype des fonctions à implémenter. *Remarque*: le paramètre K est passé explicitement à HYBRIDSORT pour faciliter vos expériences.
- `Sort.c`: le fichier à compléter. Seul l'INSERTIONSORT est fourni.
- `main.c`: un petit fichier de test.

Vos fichiers seront évalués avec la commande:

```
gcc main.c Array.c Sort.c --std=c99 --pedantic -Wall -Wextra -Wmissing-prototypes -lm -o test
```

sur les machines `ms8xx`. Ceci implique que:

- Le projet doit être réalisé dans le standard C99.
- La présence de *warnings* impactera négativement la cote finale.
- Un projet qui ne compile pas avec cette commande sur ces machines recevra une cote nulle (pour la partie code du projet).

Un projet non rendu à temps recevra une cote globale nulle. En cas de plagiat avéré, l'étudiant se verra affecter une cote nulle à l'ensemble des projets.

Les critères de correction seront précisés sur la page web des projets.

Bon travail !