

TP1 : trois algorithmes de pointage de paiements

L'objectif de ce TP est de mesurer le nombre d'opérations réalisées dans trois algorithmes dont le but est d'effectuer un pointage entre un ensemble de factures en attente de paiement et une liste de chèques de règlement. Au dos de chaque chèque est inscrit le numéro de facture. On supposera qu'il y a p chèques et n factures, que le nombre de chèques est inférieur ou égal au nombre de factures et qu'il n'existe pas de chèque ayant un mauvais numéro de facture. On souhaite produire la liste des factures **impayées**.

Vous apprendrez à générer des tests, introduire des compteurs de nombre d'opération et afficher le résultat de ces décomptes grâce à Gnuplot.

Récupérez auparavant l'archive disponible sur le portail.

Vous trouverez un répertoire `tp1` qui contient un module `Facturesetcheques`. Ce module dispose de fonctions permettant de générer des tableaux de factures et de chèques. Vous trouverez aussi un module `Tris` qui implémente une fonction de tri de tableaux utilisant le tri par insertion dichotomique ainsi qu'un programme de test et un Makefile. Vous obtiendrez le programme de test `testtp1` grâce à la commande `make` (vous n'avez pas à lancer l'interpréteur `ocaml`). Ce programme prend deux arguments : le nombre de factures et le nombre de chèques.

1 Sans prétraitement des données

Q 1 Proposez une première solution où ni les factures ni les chèques ne sont triés. Ecrivez le code de la fonction `liste_des_factures_impayees_1` de type `facture array -> cheque array -> facture list` retournant la liste des factures non réglées. Testez.

Q 2 Quelle opération va-t-on compter ? (désignons par `OP` cette opération)

Q 3 Quel est le pire des cas pour cet algorithme ?

Q 4 Exprimez $c_1(n, p)$ le nombre d'opérations `OP` pour le pire des cas en fonction de n et p .

2 Prétraitement des factures ou des chèques

Q 5 Proposez une seconde solution où les factures **OU** les chèques, mais pas les deux, sont triés (parmi les deux possibilités, choisir la plus intéressante). Ecrivez le code de la fonction `liste_des_factures_impayees_2`. Tester.

Q 6 Donnez le pire des cas et trouver une borne supérieure pour le nombre d'opérations $c_2(n, p)$, en fonction de n et p .

3 Prétraitement des factures et des chèques

Q 7 Proposez une troisième solution où les factures **ET** les chèques, sont triés. Ecrivez le code de la fonction `liste_des_factures_impayees_3`. Testez.

Q 8 Donnez le pire des cas et trouver une borne supérieure pour le nombre d'opérations $c_3(n, p)$, en fonction de n et p .

4 Recherche empirique des cas favorables

Q 9 Modifiez vos codes pour que l'on puisse compter le nombre d'opérations `OP` effectuées dans chacune des trois versions (réfléchissez bien, il n'y a quasiment rien à faire).

Q 10 Essayez de déterminer, pour chaque version, un exemplaire du problème (les valeurs de n et p suffiront) pour lequel une des versions est plus performante. Que constatez-vous ?

Q 11 Modifiez le programme de test pour qu'il permette de produire pour un nombre de factures donné en argument et un nombre de chèques variant de 1 au nombre de factures, le décompte des opérations pour les 3 versions. Pour $n = 10$, votre programme produira une sortie du type

```
10 1 10 10 39
10 2 20 17 37
10 3 30 22 39
10 4 40 28 46
10 5 50 33 48
10 6 60 39 55
10 7 70 44 59
10 8 80 50 64
10 9 90 52 66
10 10 100 61 74
```

où les colonnes correspondent respectivement à n , p et aux décomptes.

Q 12 Créez 10 fichiers pour $n = 10, 20, 30, 40, 50, 60, 70, 80, 90, 100$.

Q 13 Utilisez Gnuplot pour représenter ces données. Si vous avez redirigé la sortie de votre programme vers un fichier nommé `tp1-10.dat` pour $n = 10$ alors vous pourrez visualiser les résultats avec les commandes suivantes :

```
set title '10 factures'
plot 'tp1-10.dat' using 2:3 title 'Algo 1' w l, '' using 2:4 title 'Algo 2' w l, '' using 2:5 title 'Algo 3'
```

La suite 2:3 indique que les points en seconde colonne sont les x (p) et ceux de la troisième les y (p). `w l` (with lines) indique de relier les points par une ligne. Produisez un graphique pour chaque valeur de n . Quelle conclusion en tirez-vous ?

Note: Pour sauvegarder une image sous Gnuplot, avant de taper la commande `plot` il suffit de taper les commandes suivantes :

```
set term png
set output 'tp1-10.png'
```

pour que le graphique soit sauvegardé dans le fichier `tp1-10.png` au format PNG.