Travail pour l'évaluation du bloc 2, algorithmique

Voici un jeu d'exercices pour mettre en pratique quelques éléments que nous avons abordés dans le bloc 2 (algorithmique).

Ici, il s'agit de réaliser des programmes qui implémentent l'algorithme du tri à bulles et une amélioration du principe de base de ce tri. Nous avons vu le principe du tri à bulle en cours mais nous ne l'avons pas étudié en détail. Dans cette série d'exercices, nous reprenons cet algorithme, en donnons une version optimisée, et programmons les deux algorithmes établis.

Travail demandé

Le travail demandé est le suivant :

— rédiger la réponse à l'exercice 1 ;

— programmer cet algorithme, dans le langage de votre choix ;

— expérimenter votre programme sur des jeux de données afin de mesurer expérimentalement la complexité moyenne du nombre d'échanges ; pour cela, votre programme doit compter le nombre d'échanges de valeurs effectués dans le tableau, pour des grands tableaux dont la taille croît au fur et à mesure des expérimentations de manière à avoir des nombres d'échanges significatifs ; par exemple, vous pouvez partir de tableaux de 10000 valeurs tirées au hasard, calculer ce nombre d'échanges pour plusieurs tests, en faire la moyenne, puis doubler la taille des tableaux et reprendre les mêmes calculs ; tant qu'on y est, vous pouvez aussi mesurer les temps de calcul pour vérifier expérimentalement leur évolution ;

— rédiger la réponse à l'exercice 3 ;

— le programmer et le tester dans les mêmes conditions que précédemment ;

— l'exercice 2 **n'est pas à résoudre** dans le cadre de ce travail d'évaluation, il est ici seulement pour donner quelques indications pour le calcul de complexité moyenne du tri à bulles (en nombre d'échanges) pour ceux d'entre vous qui en ont envie ;

— pour votre information la réponse à l'exercice 2 indique que la complexité moyenne en nombre d'échanges de l'algorithme de l'exercice 1 est $n(n-1)/4$ (ce qui semble intuitivement assez naturel[1]).

Format de remise du travail

Sur deux pages, vous rédigerez les réponses aux exercices 1 et 3, en précisant autant que possible les éléments de preuves d'algorithmes (on en a déjà vu une partie en séance) ; sur une troisième page, vous donnerez les programmes de tests, et sur la dernière page (la quatrième, si j'ai bien compté), vous donnerez les résultats expérimentaux. Toutes ces pages seront regroupées dans un même fichier au format PDF. La troisième page pourra, le cas échéant, être plus longue que les autres...

Ce travail est à remettre au plus tard le 22 décembre 2019 dans l'espace réservé pour cela sur le site `magistere`

Exercice 1. Le tri à bulles (bubble-sort)

Voilà le principe du tri à bulles pour un tableau T indexé par $\{1, \ldots, n\}$. On parcourt le tableau T en examinant les paires d'éléments consécutifs : si les deux éléments d'une paire sont correctement ordonnés l'un par rapport à l'autre, on ne fait rien, sinon on les échange de manière à les réordonner.

a. Si l'on parcourt le tableau à trier des indices faibles vers les indices forts en appliquant le procédé précédent, quelle propriété possède la valeur située à l'indice n dans le tableau T après ce premier passage ? L'algorithme du tri par la méthode de la bulle consiste à répéter ce processus de parcours et d'échanges tant que le tableau n'est pas totalement trié. Établir et prouver l'algorithme du tri à bulles.

1. Mais il faut se méfier de l'intuition, en matière de complexité moyenne, on a parfois de grosses surprises.

`Patrice.Naudin@univ-poitiers.fr`

b. Pour cet algorithme, quel est le nombre de comparaisons d'éléments de tableau ? Quel est le nombre maximum d'échanges d'éléments de tableau ? Le nombre minimum ? Exhiber dans chacune de ces deux dernières situations des configurations du tableau les réalisant.

Exercice 2. Complexité du tri à bulles

Le but de cet exercice est d'établir, par trois méthodes de calcul différentes — le nombre moyen d'échanges effectués par l'algorithme du tri à bulles décrit dans l'exercice 1. On se base pour ce calcul de complexité sur l'algorithme opérant par parcours du tableau dans un seul sens (soit des indices faibles vers les indices forts, soit l'inverse).

a. Comme très souvent lorsqu'il s'agit de tri de tableau, on prend comme modèle du dérangement du tableau une permutation de $\{1, \ldots, n\}$ (ce qui n'est évidemment pas un modèle totalement réaliste, mais, là au moins, on peut faire quelques calculs). En considérant, précisément, que le tableau contient des nombres entiers tous distincts compris entre 1 et n, et en identifiant le tableau avec une permutation σ de $[1, n]$, comment peut-on décrire le nombre d'échanges d'éléments de tableau en termes de permutation ? Quel est le nombre maximum d'inversions d'une permutation de $[1, n]$?

Dans la suite de l'exercice, on considère que toutes les permutations — représentant les dérangements possibles du tableau — sont équiprobables.

b. Si σ est la permutation définie par le tableau T (i.e. $\sigma(i) = T_i$), on désigne par $\tilde{\sigma}$ la permutation définie par le tableau miroir de T. Quel rapport existe entre le nombre d'inversions de σ et celui de $\tilde{\sigma}$? Soient E_1 et E_2 les ensembles de permutations suivants :

$$E_1 = \{\sigma \mid \sigma(1) > \sigma(n)\} \qquad E_2 = \{\sigma \mid \sigma(1) < \sigma(n)\}$$

Quel est le rapport entre E_1 et E_2 ? En déduire le nombre moyen d'inversions dans une permutation. Conclure en ce qui concerne la complexité moyenne d'échanges du tri à bulles.

c. On note X_j la variable aléatoire, définie sur l'ensemble des permutations, dont la valeur est le nombre d'inversions de j. Quelle est l'espérance mathématique de cette variable aléatoire ? En déduire le nombre moyen d'échanges effectués par l'algorithme du tri à bulles.

d. Si l'on connaît la somme des nombres d'inversions de toutes les permutations de $[1, n-1]$, est-il possible d'en déduire la somme des nombres d'inversions de toutes les permutations de $[1, n]$? Résoudre la récurrence qui apparaît dans cette expression, et en déduire le nombre moyen d'échanges de l'algorithme.

Exercice 3. Amélioration du tri à bulles, le *shaker sort*

a. En triant effectivement quelques tableaux avec la méthode de la bulle, on s'aperçoit assez rapidement que, bien souvent, le tableau est trié alors que l'algorithme poursuit son exécution jusqu'à la fin prévue (heureusement ! si un algorithme réfléchissait, ce ne serait plus un algorithme). Voici quelques exemples de tableaux pour lesquels ce phénomène se produit : $[1, 2, 3, 5, 9, 8]$, et $[10, 12, 13, 21, 20, 18]$. Pour optimiser un peu le tri, on peut mémoriser le plus grand indice du tableau à partir duquel les échanges ne se font plus. Quelle est la propriété de la portion de tableau dont les indices sont inférieurs à cet indice particulier ? Esquisser une amélioration du tri par la méthode de la bulle, utilisant cette propriété.

b. Mais on peut encore améliorer la version précédente. Appliquer aux deux exemples suivants l'optimisation introduite à la question précédente : $[2, 3, 4, 5, 6, 7, 8, 1]$, $[8, 1, 2, 3, 4, 5, 6, 7]$. Peut-on généraliser l'optimisation de la question précédente (considérer ce qui se passerait si l'on exécutait l'algorithme du tri à bulles *à l'envers*) ?

En déduire un algorithme appliquant la méthode de la bulle, alternativement dans un sens et dans l'autre, en appliquant, de plus, l'optimisation de la question **a.**.

c. Quel est le nombre maximum d'échanges d'éléments du tableau opérés par cet algorithme, et dans quelle configuration de tableau.