UPHF - ISTV

année : 2019–2020
Licence 3^{eme} année Informatique
Épreuve 1^{ere} session - Semestre 5 - Resp. : R. Mandiau
Durée : 2h00 - code : S3INE3 - Bases de la Complexité
Documents autorisés - Calculatrices non autorisées
nombre de pages : 1

Exercice no	Quest. cours	hauteur max.	Un tri de cartes	Total
Points:	4	7	10	21

Exercice 1. (Quest. cours) **Exercice 1 : 4 pts**

L'algorithme étudié peut se caractériser par une relation de récurrence définie par : $T(n) = 2 \cdot T(\frac{n}{2}) + \Theta(n)$.

(a) **[2 points]** Déterminer sa complexité en utilisant la méthode des arbres récursifs.

(b) **[2 points]** Appliquer le théorème de la méthode générale pour en déduire sa complexité. Obtenons nous le même résultat que dans la question précédente ?

Exercice 2. (hauteur max.) **Exercice 2 : 7 pts**

Nous proposons une grille d'un jeu, **Puissance** 4 : l'objectif de ce jeu étant de placer quatre jetons de sa couleur dans une ligne horizontale, verticale ou diagonale. Toutefois les deux joueurs ne peuvent positionner le jeton que sur une seule dimension, la seconde dimension est déterminée par la force gravitationnelle de celui-ci [1]. Nous représentons la grille de jeu par un tableau à deux dimensions de taille n par m (par exemple, n lignes, m colonnes). Pour simplifier, nous considérons que ce tableau contient deux valeurs possibles : 0 (pas de jeton) et 1 (un jeton, de couleur indifférenciée). Nous cherchons à proposer des algorithmes qui déterminent la hauteur maximale dans cette grille.

(a) **[3 points]** Proposer un algorithme naïf. En déduire la complexité dans le pire des cas.

(b) **[4 points]** Proposer un algorithme plus efficace pour ce problème, et en déduire sa complexité dans le cas défavorable.

Exercice 3. (Un tri de cartes) **Exercice 3 : 10 pts**

Nous considérons un ensemble de n cartes numérotées de 1 à n. Le paquet de cartes est trié de la manière suivante :

— On parcourt le paquet jusqu'à trouver la première carte i qui n'est pas à la place i.
— On insère celle-ci à sa place : si la carte à ranger est la carte i, on la met en i-ème position, en décalant alors toutes celles qui sont au dessus d'elle.
— On recommence jusqu'à ce que le paquet soit trié

Dans cet exercice, il n'est pas demandé d'écrire un algorithme ; mais de produire un raisonnement correct pour en déterminer sa complexité.

(a) **[2 points]** Donner la suite des étapes successives du tri d'un paquet de 5 cartes initialement dans l'ordre $(3, 2, 5, 4, 1)$.

(b) **[2 points]** Déduire la complexité en nombre de déplacements de cartes (considéré ici comme l'opération fondamentale) dans le cas défavorable.

(c) **[3 points]** Supposons maintenant que nous associons à chaque ordre possible du paquet un poids $p = \Sigma_{i=1}^{n} 2^{i-1} d_i$ où $d_i = 1$ si la carte i est à sa place dans le paquet et 0 sinon. Par exemple $p(1, 2, 4, 3, 5) = 1 + 2 + 16 = 19$ car 1, 2 et 5 sont à leur place (3 et 4 ne le sont pas). Montrer qu'à chaque fois que nous déplaçons une carte selon l'algorithme considéré le poids de l'ordre du paquet augmente strictement.

(d) **[3 points]** En déduire une preuve que l'algorithme termine toujours et donner une majoration de sa complexité dans le pire des cas (en nombre de déplacements de cartes).

1. Il n'est pas nécessaire de connaître le jeu pour traiter cet exercice.