EPFL - Information, Calcul et Communication - Partie théorique

Exercices additionnels et solutions

Cinq algorithmes à écrire

Après avoir résolu chaque problème ci-dessous, vous êtes invités à réfléchir si la solution que vous avez trouvée est la plus efficace en temps de calcul. La meilleure (i.e., la plus petite) complexité temporelle qu'il est possible d'obtenir pour chaque algorithme est donnée au bas de la page.

a) Soit L une liste de n nombre entiers. Ecrire un algorithme qui trouve la plus *grande* différence en valeur absolue entre deux nombres de la liste.

Exemple: Si $L = (-3, 17, 15, -22, 3)$ et $n = 5$, alors la sortie doit être $39 = |17 - (-22)|$.

b) Soit L une liste de n nombre entiers. Ecrire un algorithme qui trouve la plus *petite* différence entre deux nombres de la liste.

Exemple: Si $L = (-3, 17, 15, -22, 3)$ et $n = 5$, alors la sortie doit être $2 = |17 - 15|$.

c) Supposons qu'on dispose d'un algorithme **appartient à**(L, k) dont la sortie soit oui si la taille de la liste L est plus grande ou égale à k (i.e., si $L(k)$ est un élément appartenant à la liste L), et non dans le cas contraire. Ecrire un autre algorithme utilisant celui-ci qui calcule la taille de la liste L.

Exemple: Si $L = (-3, 17, 15, -22, 3)$, alors la sortie doit être simplement $n = 5$.

NB: On suppose ici que la complexité temporelle de l'algorithme **appartient à**(L, k) est $\Theta(1)$.

d) Supposons qu'on dispose d'un algorithme **aléatoire**(n) qui tire un nombre entier uniformément au hasard entre 1 et n. Ecrire un autre algorithme utilisant celui-ci qui à partir d'une liste ordonnée L de n nombres entiers, génère une liste aléatoire A qui soit une version "dans le désordre" de la liste L (donc tous les nombres de L doivent se retrouver dans la liste A).

Exemple: Si $L = (-22, -3, 3, 15, 17)$ et $n = 5$, alors la sortie peut être par exemple $A = (15, 3, 17, -22, -3)$.

NB: On suppose ici que la complexité temporelle de l'algorithme **aléatoire**(n) est $\Theta(n)$.

e) Soit n un nombre entier positif et A un tableau de dimensions $2 \times n$ rempli de nombres entiers positifs tels que *dans chacune des 2 lignes du tableau, tous les nombres sont différents.* Par exemple, si $n = 7$, un tel tableau pourrait être:

$$A = \begin{pmatrix} 4 & 3 & 11 & 2 & 12 & 1 & 5 \\ 5 & 2 & 12 & 13 & 15 & 4 & 3 \end{pmatrix}$$

Ecrire un algorithme dont les entrées soient un tableau A de ce type et n le nombre de colonnes de A, et dont la sortie soit oui s'il existe un numéro de colonne $i_0 \in \{1, \ldots, n\}$ tel que

$$A(1, i_0) \geq A(1, i) \quad \text{et} \quad A(2, i_0) \geq A(2, i) \quad \text{pour tout } i \in \{1, \ldots, n\}$$

et non dans le cas contraire (dans l'exemple ci-dessus, la sortie doit donc être oui).

a) $\Theta(n)$ b) $\Theta(n \cdot \log_2(n))$ c) $\Theta(\log_2(n))$ d) $\Theta(n^2)$ e) $\Theta(n)$

Exercice 1. On considère les deux algorithmes suivants:

machin

entrée : *nombre entier positif* n
sortie : *???*

Si $n = 1$
  Sortir: 1

Sortir : $2^{(\textbf{bidule}(n-1)+1)}$

bidule

entrée : *nombre entier positif* n
sortie : *???*

Si $n = 1$
  Sortir: 0

Sortir : $\log_2(\textbf{machin}(n-1))$

a) Quelle est la sortie de l'algorithme **bidule** lorsque $n = 4$ en entrée?

b) Quelle est la complexité temporelle de l'algorithme **bidule**? (utiliser la notation $\Theta(\cdot)$)

c) Réécrire l'algorithme **machin** de façon récursive, mais sans faire appel à **bidule**.

d) En déduire quelle est la sortie de **machin** pour une valeur quelconque de $n \geq 1$ en entrée.

Exercice 2. Un signal sonore $X = (X(t),\, t \in \mathbb{R})$ de bande passante $B = 6$ kHz est d'abord filtré par un filtre passe-bas idéal de fréquence de coupure f_c avant d'être échantillonné à une fréquence d'échantilonnage f_e, et chaque échantillon est ensuite encodé sur 32 bits.

a) Supposons que l'on souhaite enregistrer 50 minutes de ce signal sur un support mémoire d'une taille de 120 Mégaoctets (un octet représentant 8 bits). Quelles fréquences f_c et f_e choisir pour conserver au maximum les fréquences du signal X sans pour autant subir l'effet stroboscopique lors de la reconstruction du signal?

b) Si maintenant le signal X est donné par la formule explicite suivante:

$$X(t) = \sum_{n \geq 1} \frac{1}{n} \sin\left(\frac{2\pi f_0 t}{n}\right), \quad t \in \mathbb{R}$$

où $f_0 = 6$ kHz, quelle sera la formule pour le signal reconstruit Y dans le scénario précédent?

Exercice 3. Un texte est composé pour moitié de lettres tirées d'un alphabet de 16 lettres, et pour moitié d'espaces. On suppose de plus que les probabilités d'apparition de toutes les lettres sont égales, que les lettres et les espaces sont mélangés dans le texte, et que plusieurs espaces peuvent être consécutifs.

a) Proposez un système d'encodage de ce texte sous la forme d'une séquence de bits qui permette d'utiliser le plus petit nombre moyen de bits par caractère (lettre ou espace), tout en restant décodable de manière unique au moyen d'un dictionnaire; que vaut ce nombre?

b) Si maintenant le texte était composé pour deux tiers de lettres et pour un tiers d'espaces, l'encodage proposé en a) serait-il toujours optimal? Calculez également le nombre moyen de bits utilisés par caractère dans ce cas.

Exercice 4. Afin d'envoyer un message sur internet, Alice décompose d'abord celui-ci en deux paquets de données X_1, X_2, composés chacun d'une séquence de 1024 bits. Alice envoie ensuite X_1 et X_2, ainsi que deux copies $X_3 = X_1$ et $X_4 = X_2$ de ces paquets, et également un cinquième paquet $X_5 = X_1 \oplus X_2$ (ce qui signifie que chaque bit de X_5 est l'addition modulo 2 des bits des paquets X_1 et X_2).

a) Bob est le destinataire de ces paquets, mais malheureusement, certains d'entre eux se perdent en cours de route... Combien de pertes de paquets Bob peut-il tolérer, dans le pire des cas, tout en restant sûr de pouvoir retrouver le message orignal X_1, X_2 envoyé par Alice? Expliquez votre raisonnement.

b) Supposons maintenant qu'Alice et Bob se mettent d'accord à l'avance sur une clé secrète K, dont les bits sont tirés uniformément au hasard. Expliquez comment Alice et Bob peuvent utiliser cette clé pour échanger les cinq paquets décrits ci-dessus, de sorte que si Eve intercepte même tous les paquets, elle ne puisse retrouver aucune information à propos de X_1 ou X_2. En particulier, quelle doit être la longueur minimum de la clé K (en nombre de bits)?

Solutions des exercices

Exercice 1. a) la sortie vaut 1 dans ce cas.

b) A chaque étape, l'algorithme **bidule** ou **machin** effectue $\Theta(1)$ opérations et appelle l'autre algorithme avec une valeur de n décrémentée de 1, jusqu'à la condition de terminaison où $n = 1$. La complexité est donc $\Theta(n)$.

c)

machin_machin

entrée : *nombre entier positif n*
sortie : *nombre entier positif s*

Si $n = 1$ *ou* 2
  Sortir: n

Sortir : $2*$**machin_machin**$(n-2)$

d) Si n est pair, $s = 2^{n/2}$. Si n est impair, $s = 2^{(n-1)/2}$. En général: $s = 2^{\lfloor n/2 \rfloor}$.

Exercice 2. a) On a l'égalité $120 \times 10^6 \times 8 = 50 \times 60 \times f_e \times 32$. En simplifiant, on trouve $2 \times 10^6 = 50 \times f_e \times 4$, donc $f_e = 10$ kHz. Il faut donc choisir $f_c < 5$ kHz pour respecter la condition de Nyquist.

b) Les fréquences de ce signal sont $f_0 = 6$ kHz, $f_0/2 = 3$ kHz, $f_0/3 = 2$ kHz, etc. Avec une fréquence de coupure juste un peu plus petite que 5 kHz, le signal sortant (qui ne subit pas l'effet stroboscopique) est donc:

$$X(t) = \sum_{n \geq 2} \frac{1}{n} \sin\left(\frac{2\pi f_0 t}{n}\right), \quad t \in \mathbb{R}$$

Exercice 3. a) La réponse est indépendante de la longueur du texte, car seules les probabilités d'apparition des lettres comptent. On peut supposer par exemple que chaque lettre apparaît une fois et que l'espace apparaît 16 fois. En créant l'arbre de Huffman, on trouve que l'espace est représenté par 1 bit et que chaque lettre est représentée par 5 bits, donc en moyenne: $\frac{1}{2}\,1 + \frac{1}{2}\,5 = 3$ bits par caractère.

b) Dans ce cas, on peut supposer que chaque lettre apparaît 2 fois et que l'espace apparaît 16 fois. En reconstruisant l'arbre de Huffman dans ce cas, on voit que la représentation prcécédente utilisant 1 bit pour l'espace et 5 bits pour les autres lettres est toujours optimale. On obtient alors en moyenne $\frac{1}{3}\,1 + \frac{2}{3}\,5 = \frac{11}{3} \simeq 3,67$ bits par caractère.

Exercice 4. a) Si on pense aux paquets X_1, X_2, X_3, X_4, X_5 comme à des bits isolés, on voit que les mots de code possiblement envoyés sont:

$$00000 \qquad 10101 \qquad 01011 \qquad \text{et} \qquad 11110$$

La distance minimale de ce code correcteur d'erreur vaut $d = 3$: il est donc possible de corriger 2 effacements, dans le pire des cas.

Note: Dans certains cas, il est possible de corriger jusqu'à 3 effacements (si par exemple les paquets X_3, X_4, X_5 sont effacés), mais pas dans le pire des cas.

b) Il suffit que la clé K soit d'une longueur de 2048 bits. En effet, divisons cette clé en deux parties égales K_1 et K_2. Alice calcule $Y_1 = X_1 \oplus K_1$ et $Y_2 = X_2 \oplus K_2$ et transmet:

$$Y_1, Y_2, Y_1, Y_2, Y_1 \oplus Y_2$$

Le même niveau de correction d'erreurs est ainsi assuré, et même si Eve intercepte tous les paquets, elle ne peut déchiffrer ni X_1, ni X_2.