

Série 4

1 Une meilleure version du tri par fusion?

Au cours, nous avons vu le tri par fusion, dont la complexité temporelle est $\Theta(n \cdot \log_2(n))$.

Voici maintenant une proposition d'amélioration de cet algorithme: au lieu de diviser la liste à trier en deux parties à chaque étape, divisons-la en quatre parties, trions chacune de ces quatre parties séparément (au moyen de notre nouvel algorithme), et fusionnons ensuite ces quatre parties en une seule. Quelle sera alors la complexité temporelle de ce nouvel algorithme, si on suppose que l'algorithme de fusion de quatre listes L_1, L_2, L_3, L_4 a une complexité temporelle $\Theta(m_1 + m_2 + m_3 + m_4)$, où m_i est la taille de la liste L_i ?

Question subsidiaire: Et si on pousse l'idée encore plus loin et qu'on divise la liste en une seule fois en n unités au départ, puis qu'on fusionne ensuite ces n unités également en une fois, quelle complexité temporelle obtient-on?

2 Nombres de Fibonacci

La suite des nombres de Fibonacci est définie de façon récursive comme suit:

$$F(1) = F(2) = 1, \quad F(n) = F(n-1) + F(n-2), \quad \text{pour } n \geq 3$$

(et vaut donc 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, ...)

- Ecrivez un algorithme récursif dont l'entrée soit un nombre entier positif n et dont la sortie soit $F(n)$.
- Quelle est la complexité temporelle de votre algorithme? (utilisez la notation $\Theta(\cdot)$)
- Pouvez-vous imaginer un autre algorithme, itératif cette fois, permettant de calculer $F(n)$ en un temps considérablement moindre?

3 Rendu de pièces de monnaie

Pour un montant à rendre valant $z = 2.60$ frs et pour chacun des ensembles de pièces de monnaie ci-dessous:

$$P_1 = (40 \text{ cts}, 70 \text{ cts}, 1 \text{ fr.})$$

$$P_2 = (50 \text{ cts}, 70 \text{ cts}, 1 \text{ fr.})$$

$$P_3 = (10 \text{ cts}, 30 \text{ cts}, 40 \text{ cts}, 1 \text{ fr.})$$

$$P_4 = (20 \text{ cts}, 40 \text{ cts}, 1 \text{ fr.})$$

déterminez si l'algorithme de rendu "glouton" vu au cours:

- rend le montant exact;
- utilise le nombre minimum de pièces pour cela.

Dans les cas où la réponse à l'une des deux questions ci-dessus est négative, déterminez quelle est la solution optimale (si une telle solution existe).

4 Un autre algorithme mystère

Quelle est la sortie de l'algorithme suivant si la donnée en entrée est un nombre entier positif $n \leq 100$?

algo
entrée : <i>nombre entier positif</i> n sortie : <i>nombre entier positif</i> s
Si $n > 100$ Sortir: $n - 10$ Si non Sortir: algo (algo ($n + 11$))

Indication: L'étude de la sortie d'un tel algorithme peut vite donner le tournis! Posez-vous tout d'abord la question: pour quelles valeurs de n en entrée cet algorithme s'arrête-t-il? Regardez ensuite ce qui se passe pour des nombres n entre 90 et 100.

5 Pour le plaisir: deviner une date d'anniversaire* (©FSJM, 2017)

Anne donne à Manon et à Julie dix dates possibles pour son anniversaire :

15, 16 et 19 mai; 17 et 18 juin; 14 et 16 juillet; 14, 15 et 17 août

Elle donne le jour (un nombre de 14 à 19) à Julie, mais pas à Manon, et le mois à Manon, mais pas à Julie.

Manon dit à Julie: "Je ne sais pas quelle est la date, mais je sais que tu ne le sais pas non plus".

Julie répond à Manon: "Je ne savais pas quelle était la date, mais maintenant je le sais".

Manon conclut: "Alors je sais aussi quelle est la date".

Quelle est la date de l'anniversaire d'Anne ?