

Algorithmique III (I41) - Licence d'Informatique

Contrôle terminal (Session 2 - Juillet 2021)

En 2022, les notes d'algorithmique au contrôle terminal sont remplacées par quatre smileys ☹, 😊, 😄 et 😈 quand la copie est vraiment drôle, et sont encodés par les entiers naturels 1, 2, 3 et 0, *respectivement*. On note n le nombre d'étudiants, E la liste de leurs noms et T la liste des codes des notes correspondantes ($T[i]$ est la note de l'étudiant $E[i]$ pour tout $i \in [1, n]$).

(1) Zack a eu pour résultat ☹, Liza 😊, Phil 😊, Eva 😊 et Liz 😊. Quelles sont les listes E et T pour ces 5 étudiants si E a été triée dans l'ordre lexicographique et T en conséquence afin que $T[i]$ soit la note de $E[i]$ pour tout $i \in [1, n]$?

(2) Quelle relation d'ordre total \preceq faut-il définir sur l'ensemble

$$\mathcal{S} := \{\ominus, \omin�, \omin�, \omin�\}$$

pour que la fonction d'encodage $e : \mathcal{S} \rightarrow \{0, 1, 2, 3\}$ telle qu'elle a été définie en introduction soit strictement croissante si l'on munit $\{0, 1, 2, 3\}$ de l'ordre naturel \leq ? Justifiez.

(3) Combien existe-t-il de listes distinctes de notes de longueur n possibles ? Justifiez.

(4) Écrivez un algorithme `TrierNotes(@T)` qui trie la liste des notes T des n étudiants dans l'ordre croissant. Ce tri ne devra faire *aucune* comparaison entre notes de la liste T .

(5) Quelle est la complexité de votre algorithme ? Justifiez.

(6) Peut-on adapter l'algorithme `TrierNotes` de manière à trier simultanément la liste E afin que $T[i]$ soit la note de l'étudiant $E[i]$ pour tout $i \in [1, n]$, sans changer sa complexité ? Justifiez.

Après l'examen, les copies ont été corrigées et la liste T triée par l'algorithme `TrierNotes`. Les étudiants les moins bien notés décident de se plaindre parce qu'ils estiment que l'examen était trop dur. La promotion est alors partitionnée en deux groupes : le groupe des frondeurs aux positions 1 à q , et les autres aux positions $q + 1$ à n avec $q < n$. La césure q a été réalisée de manière à ce qu'il y ait le plus de frondeurs possible, mais sans que la somme de leurs notes n'excède celle des notes des autres étudiants.

(7) Quelle est la valeur de q pour la liste des notes de la question (1) ? Justifiez.

(8) Donnez une définition formelle de l'entier q à l'aide d'une expression en logique des prédicats.

(9) Écrivez un algorithme `Frondeurs(T)` qui calcule et renvoie la valeur q , en supposant que la liste des notes T a été triée au préalable en appelant `TrierNotes(@T)`.

(10) Faites une preuve d'arrêt de votre algorithme et justifiez qu'il renvoie bien la valeur q .

(11) Calculez la complexité de votre algorithme dans le meilleur des cas et dans le pire des cas. Justifiez.