

Examen d'Algorithmique II

Durée : 1h30

Exercice 1 : Analyse de problèmes (5 points)

1. Problème 1. Somme de chiffres (1 points)

Réaliser la fonction `addDigits` qui calcule, pour un entier, la somme des chiffres de cet entier.

`addDigits` : fonction(N : entier ≥ 0) retourne entier ≥ 0

{ `addDigits(N)` désigne la somme des chiffres constituant l'entier N }

Exemples : `addDigits(8263643) = 32`

`addDigits(25) = 7`

2. Problème 2. Supprimer les doubles dans une séquence (4 points)

On considère une séquence d'entiers triés en ordre croissant, S de longueur N représentée dans un tableau T d'entiers défini sur l'intervalle $[1 \dots N_{\max}]$, $0 \leq N \leq N_{\max}$.

On supposera que le tableau considéré contient déjà la séquence, sa saisie dans le tableau n'est donc pas à faire.

On veut écrire un algorithme qui remplace dans T la suite S par la suite S' de longueur N' (avec $N' \leq N$), déduite de S en supprimant tous les éléments redondants. Un élément de S est redondant s'il est égal à un autre élément de S .

L'algorithme ne doit pas utiliser de tableau intermédiaire pour créer S' .

L'ordre des éléments reste celui de la séquence de départ.

Travail demandé : faire l'analyse descendante du problème dans l'hypothèse où T est trié en ordre croissant.

Exemple : si $S = [33, 54, 54, 98, 101, 101, 150, 169, 169]$ et $N = 10$, alors $S' = [33, 54, 98, 101, 150, 169]$ et $N' = 6$.

Exercice 2: tris et complexité (5 points)

1. Donner l'algorithme de tri par insertion. (2 points)

2. Exprimer sa complexité dans le cas pire en fonction de la taille du tableau. (3 points)

Exercice 3 : Comparaison d'algorithmes de tri (5 points)

On considère le tableau d'entiers suivants : $T = [0, 8, 10, 2, 9, 4, 4]$.

1. Appliquer l'algorithme de tri à bulles sur le tableau T et donner sa trace d'exécution. (2 points)
2. Même question, en appliquant cette fois-ci l'algorithme de tri par sélection. (2 points)
3. Comparer les deux exécutions en termes de nombre d'échanges effectués? Et conclure. (1 point)

Exercice 4 : Preuve d'algorithme (5 points)

1. Transformer une boucle *Pour* : *Pour i allant de E jusqu'à F Faire A Fin_Pour* en une boucle *Tantque*. (1 point)
2. En utilisant cette transformation de la boucle *Pour* à l'aide de *Tantque*, écrire la règle de Hoare correspondant à la construction : *Pour i allant de E jusqu'à F Faire A Fin_Pour*. (2 points)
3. Ecrire un algorithme itératif naïf du calcul du factoriel d'un nombre entier positif et prouver sa correction en utilisant les règles de Hoare. (2 points)