

Examen Semestriel

Durée 1^h00^m Date : 28/03/2021

Algorithmique Avancée et Complexité

Exercice 1 (6 points)

Soit les trois parties d'Algorithme suivantes

Partie 1	Partie 2	Partie 3
Pour i :=1 à n <u>faire</u> Pour j := 1 à i <u>faire</u> Ecrire ('Hello') <u>fait</u> <u>fait</u>	Pour i :=10 à n -10 <u>faire</u> Pour j := i-10 à i+10 <u>faire</u> Ecrire ('Hello') <u>fait</u> <u>fait</u>	i :=n Tant que i>1 <u>faire</u> pour j := 0 à i <u>faire</u> ecrire('hello') <u>fait</u> ; i := [i /2] ; <u>fait</u>

Déterminer en fonction de n la complexité des tris parties en terme d'ordre de grandeur asymptotique

Exercice 2 (6 points)

On souhaite enregistrer sur une mémoire de taille L un groupe de fichiers $P = (P_1, \dots, P_n)$. Chaque fichier P_i nécessite une place a_i . Supposons que $\sum a_i > L$: on ne peut pas enregistrer tous les fichiers. Il s'agit donc de choisir le sous ensemble Q des fichiers à enregistrer.

- 1- Écrivez un algorithme glouton qui permettra de construire le sous ensemble Q, tel que le nombre de fichier soit maximal. Cet algorithme doit renvoyer un tableau booléen S tel que $S[i] = 1$ si P_i est dans Q et $S[i] = 0$ sinon.
- 2- Evaluer sa complexité temporelle ?

Exercice 3 : (7 Points)

Étant donné un tableau T de n entiers et un entier x, on veut déterminer s'il existe deux éléments de T dont la somme est égale à x.

- 1- Donner un algorithme le plus simple possible, basé sur la comparaison, Evaluer ça complexité en pire des cas (Justifier) ?
- 2- Donner un algorithme en $O(n \log n)$ en pire cas ? (Justifier) -Vous pouvez utiliser des algorithmes vus en cours et les résultats de complexité sur ces algorithmes-