

	ANNÉE UNIVERSITAIRE 2016/2017 SESSION 2 D'AUTOMNE CPBX AN2	Collège Sciences et Technologie
	Parcours / Étape : ...      Code UE : 4TBX311 Épreuve : 4TBX311U Informatique Date : 24/03/2017    Heure : 14h00    Durée : 1h30 Documents non autorisés Épreuve de M. Dorbec Paul	

# 1 Recherche de deux éléments de somme fixée.

## 1.1 Première approche

On cherche à écrire un algorithme qui prend en entrée deux listes d'entiers  $A$  et  $B$  et un entier  $x$ , et qui cherche s'il existe un élément de  $A$  et un élément de  $B$  qui ont pour somme  $x$ . On notera  $a$  le nombre d'éléments de la liste  $A$  et  $b$  le nombre d'éléments de  $B$ .

La première idée qui vient à l'esprit consiste à regarder pour chaque élément de  $A$  et pour chaque élément de  $B$  si leur somme vaut  $x$ .

1. Écrivez le pseudo-code de cet algorithme naïf, retournant vrai s'il existe deux éléments dont la somme vaut  $x$ , faux sinon.
2. Quelle est la complexité de cet algorithme ?

## 1.2 Et si les listes sont triées ?

On a l'intuition qu'il pourrait être utile de trier les listes  $A$  et  $B$  avant de chercher à résoudre le problème.

3. Donnez un exemple d'algorithme de tri ayant une bonne complexité dans le pire cas ? Quelle est cette complexité ?
4. Expliquer comment utiliser le fait que les listes sont triées pour résoudre le problème. Vous pourrez écrire le pseudo code d'un algorithme résolvant le problème sur ces listes triées.
5. Quelle est la complexité de la méthode de recherche dans les listes triées que vous avez décrite ?
6. Quelle est la complexité dans le pire cas de l'algorithme consistant à trier les listes puis à faire la recherche décrite ci-dessus ?

### 1.3 Diviser pour régner

Nous allons maintenant mettre en place une approche *diviser pour régner* de la question. On ne suppose plus que les listes sont triées.

La première étape consiste à couper chacune des deux listes en deux :  $A_1$  contiendra les éléments de  $A$  plus petits que  $\frac{x}{2}$  et  $A_2$  les éléments plus grands ou égaux à  $\frac{x}{2}$ .

7. Quelle est la complexité de l'opération consistant à créer  $A_1$  et  $A_2$  à partir de  $A$  ?
8. Expliquez comment utiliser ce découpage des listes  $A_1$ ,  $A_2$ ,  $B_1$ , et  $B_2$  pour résoudre le problème.

Le problème de cette opération, c'est qu'on peut difficilement la répéter sur les listes déjà découpées. Une idée consiste à remplacer tous les éléments  $a_i$  de la liste  $A_2$  par  $a_i - \frac{x}{2}$  et de même sur la liste  $B_2$ .

9. Comment adapter la solution précédente aux nouvelles listes  $A_1$ ,  $A_2$ ,  $B_1$ , et  $B_2$  ainsi modifiées ?
10. Écrivez le pseudo-code de ce nouvel algorithme.

On note  $C(a, b)$  la complexité de vérifier si deux éléments de listes  $A$  et  $B$  de tailles respectives  $n$  et  $m$  ont pour somme une valeur  $x$  avec l'algorithme ci-dessus. On note  $a_1$  la taille de la liste  $A_1$  et  $b_1$  celle de la liste  $B_1$ .

11. Quelle relation de récurrence vérifie  $C(a, b)$  ?
12. Supposons que les découpages des listes sont équilibrés (*i.e.*  $a_1 = \frac{a}{2}$  et  $b_1 = \frac{b}{2}$ ). Que donne cette formule de récurrence ?
13. Que donne-t-elle si on l'applique deux fois ? trois fois ?  $k$  fois ?
14. Intuïtez la complexité de cet algorithme dans ce cas.