

# Série analyse amortie

## 1 Notes

Même si une trace d'exécution n'est pas demandé, il est recommandé de faire un exemple de séquences d'opérations avant de faire une analyse amortie.

## 2 File à deux piles

Soit une file  $f$ , elle peut être représentée de la manière suivante :

1. *Entrants* : une pile pour les éléments qui entrent lors d'une insertion
2. *Sortants* : une autre pile dédiée aux éléments qui vont sortir pour le dépilement. Dans le cas où elle est vide et que *Entrants* ne l'est pas, alors le contenu de l'autre pile est transféré dans *Sortants* pour après enlever l'élément en tête de cette pile dans le cas où on dépile un élément dans la file.

Répondez à la question suivante :

1. Faites l'analyse amortie des différentes opérations avec la méthode de l'agrégat, du comptable et du potentiel.

## 3 Dictionnaire de tableaux

Une manière de représenter un "dictionnaire" est la suivante : on a ensemble de tableaux qui ont une taille équivalente à une puissance de 2. Ces derniers sont soit vide ou pleine. Dans le cas où elles sont pleines, elles sont triées. En revanche, malgré qu'il y ait plusieurs tableaux triés, ces dernières ont aucune relation entre elles.

Voici un exemple avec 11 éléments dans notre "dictionnaire" :

$R_1$	[6]
$R_2$	[2, 10]
$R_3$	<i>Vide</i>
$R_4$	[-1, 2, 5, 7, 8, 9, 10, 25]

Pour savoir quels sont les tableaux qui sont pleins, il faut regarder la représentation binaire du nombre d'élément présents. Ici,

$$(11)_{10} = (1011)_2$$

les puissances de 2 qui sont marqués à 1 dans la representation binaire du nombre d'éléments. Ici, les puissances 0, 1, 3 signifient qu'il y a un tableau plein des tailles suivantes : 1, 2, 8. Lors de l'insertion d'un nouvel élément, les deux situations suivantes peuvent se produire :

1. S'il n'y a pas de tableau de taille 1 qui est déjà présente, alors on crée seulement un tableau de taille 1.
2. Dans le cas où un tableau de taille 1 est déjà présent, alors on fusionne ces deux tableaux en les mettant les éléments dans l'ordre croissant. Dans le cas de la fusion de deux tableaux triés (de même taille dans ce cas-ci), il est possible de les fusionner en  $\Theta(p)$  où  $p$  est le nombre d'éléments des deux tableaux. Ensuite, dû à ce que le nouveau tableau a doublé en taille, il faut vérifier s'il existe un tableau de même taille que le nouveau tableau. Si c'est le cas, on répète le même processus jusqu'à ce que le tableau créé d'une taille  $2^c$  où  $c$  est une constante soit le seul tableau de cette taille.

Par exemple, si on ajoute l'élément 3 dans la structure de tout à l'heure, on obtient le résultat suivant :

$R_1$	<i>Vide</i>
$R_2$	<i>Vide</i>
$R_3$	[2, 3, 6, 10]
$R_4$	[-1, 2, 5, 7, 8, 9, 10, 25]

Or,  $(12)_{10} = (1100)_2$  ce qui signifie qu'il y a seulement un tableau pour les tailles suivantes : 4 et 8. Pour rechercher un élément, on peut juste effectuer une recherche dichotomique (binary search) sur chaque tableau jusqu'à ce que des tableaux contiennent l'élément ou pas.

Répondez aux questions suivantes :

1. Donner la complexité dans le pire des cas pour l'insertion d'un nouvel élément et sa complexité pour insérer un nouvel élément.
2. Donner la complexité de la recherche d'un élément dans cette structure de donnée.
3. Si on ajoute une opération pour supprimer un certain élément de la structure de donnée, proposez une approche et faites l'analyse dans le pire des cas et son analyse amortie.

## 4 Annexe

### 4.1 Exercice 1

Voici les algorithmes des opérations et d'autres attributs pour la structure de données :

1. **Procédure** ENFILER(element)
2.     empiler(element, f.entrant)
3. Fin Procédure

Note : on peut supposer que cette fonction se fait en temps constant.

1. **Fonction** ESTVIDE(P)
2.     Renvoyer Taille(P) = 0
3. Fin Fonction
  
1. **Fonction** DÉPILER()
2.     Si estVide(f.sortant) alors
3.         Renvoyer **erreur**
4.     Sinon
5.         Si Taille(f.sortants) = 0 alors
6.             Tant que  $\neg$  EstVide(f.entrants) faire
7.                  $x \leftarrow$  depiler(f.entrants)
8.                 empiler(x, f.sortants)
9.         Fin tant que
10.         Renvoyer depiler(f.sortants)
11.     Sinon

12. Renvoyer depiler(f.sortants)
13. Fin Si
14. Fin Si
15. Fin fonction