

Algorithmique 1

Feuille 8 : Tas et Files de priorité

1 Tas

Soit le type abstrait `tas` introduit en cours¹.
Pour rappel voir annexes A, B et C.

Exercice 8.1

- Dessiner tous les tas (min) possibles avec l'ensemble $\{1, 2, 3, 4, 5\}$ sachant que chaque chiffre n'apparaît qu'une seule fois.
- Combien de tas (min) peut-on construire sur un ensemble ordonné de 3, de 5 et de 7 éléments? Question difficile, peut-on généraliser à n ?

Exercice 8.2

Considérant un tas (min) dont l'ordre dépend des entiers croissant. Ecrire chaque arbre intermédiaire résultant de la suite de nombres suivante :
21 , 12 , 6 , 24 , 6 , 20 , 13 , 10 , 6.
Pour les 6 premiers appels, détaillez le déroulement de la primitive `ajouter`.

Exercice 8.3 *En utilisant l'implémentation d'un tas (min)*

- Ecrire un algorithme permettant de connaître le plus petit élément d'un tas. Quel est sa complexité?
- Ecrire un algorithme permettant de connaître le plus grand élément d'un tas. Quel est sa complexité?

Exercice 8.4 *Tri*

- Ecrire un algorithme de tri permettant d'obtenir une liste contenant les éléments par ordre croissant à partir d'un tas min.
- Faites tourner cet algorithme sur le tas engendré par l'exercice 2.
- Quelle est sa complexité?

Exercice 8.5

- Ecrire une fonction qui teste si un arbre binaire est un tas (min).
- Faites tourner cet algorithme sur le tas engendré par l'exercice 2.
- Quelle est sa complexité?

2 File de priorité

Problème récurrent (notre fil d'ariane)

Exercice 8.6 Gestion d'une piste d'atterrissage des avions

Un avion est un enregistrement contenant :

- L'indicatif (6 caractères)
- La destination (30 caractères)
- L'autonomie résiduelle de carburant comptée en heures de vol (entier)
- Deux booléens indiquant s'il y a un pirate à bord et s'il y a le feu.

1. Définir les structures de données nécessaires.
2. Ecrire la fonction `Priorité` ainsi que la gestion complète de la piste.

1. Un tas max (resp. tas min) est un arbre binaire quasi-parfait étiqueté par des objets comparables tel que tout noeud a une étiquette plus grande (resp. plus petite) que ses fils. Un tas est un conteneur et un arbre binaire, il dispose donc des primitives des arbres binaires ainsi que ceux d'un conteneur.

3. Envisager le cas de suppression d'un élément quelconque de la file lorsque le pirate a mis sa menace de détournement à exécution.

Quelles sont les notions que vous venez de voir qui peuvent permettre d'amorcer le fil d'ariane ?

ANNEXE A : Type abstrait tas.

```
fonction valeur(val T: tas d'objet): objet;
fonction ajouter(ref T: tas d'objet, val v: objet): vide;
fonction supprimer(ref T: tas d'objet): vide;
fonction creerTas(ref T: tas d'objet, val v: objet): vide;
fonction detruireTas(ref T: tas d'objet): vide;
```

ANNEXE B : Implémentation du type abstrait tas.

```
tas=structure
    arbre:tableau[1..tailleStock] d'objet;
    tailleTas:entier;
finstructure;
 curseur=entier;
 sommet=entier;
fonction getValeur(val T: tas d'objet, val s: sommet): objet;
fonction valeur(val T: tas d'objet): objet;
fonction filsGauche(val s: sommet): sommet;
fonction filsDroit(val s: sommet): sommet;
fonction pere(val s: sommet): sommet;
fonction setValeur(ref T: tas d'objet, val s: sommet, val x:objet): vide;
fonction tasPlein(val T: tas d'objet): booleen
fonction creerTas(ref T: tas d' objet, val racine: objet): vide;
fonction ajouter(ref T: tas d'objet, val v: objet): vide;
fonction supprimer(ref T: tas d'objet): vide;
```

ANNEXE C : Type abstrait file de priorité.

Primitive supplémentaire :

```
fonction changeValeur(ref T: tas d'objet, val s: sommet, val v: objet): vide;
```