

Un tas

Définition

Tamiser un élément du
tableau

Entasser le tableau

Tri par tas

Tri par tas

Christophe Viroulaud

Terminale - NSI

Algo 09

Un tas

Définition

Tamiser un élément du
tableau

Entasser le tableau

Tri par tas

Le tri rapide ou le tri fusion ont une complexité quasi-linéaire ($O(n \times \log_2 n)$).

Peut-on trier efficacement un tableau à l'aide d'un arbre binaire ?

1. Un tas

1.1 Définition

1.2 Tamiser un élément du tableau

1.3 Entasser le tableau

2. Tri par tas

Un tas

Définition

Tamiser un élément du tableau

Entasser le tableau

Tri par tas

Un **arbre partiellement ordonné** est tel que la valeur de chaque nœud fils est inférieure au nœud père (figure 8).

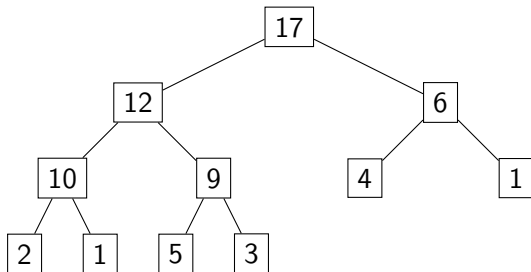


FIGURE 1 – Arbre partiellement ordonné

Un tas

Définition

Tamiser un élément du tableau

Entasser le tableau

Tri par tas

Un **tas** est un tableau dont l'arbre associé est un arbre partiellement ordonné (code 1).

```
1 tas = [17, 12, 6, 10, 9, 4, 1, 2, 1, 5, 3]
```

Code 1 – Tas associé à l'arbre 8

Chaque nœud fils est accessible en respectant la convention suivante :

- ▶ l'indice de la racine est 0,
- ▶ l'indice du fils gauche est $2 * i + 1$,
- ▶ l'indice du fils droit est $2 * i + 2$.

Un tas

Définition

Tamiser un élément du tableau

Entasser le tableau

Tri par tas

Un tas

Définition

Tamiser un élément du tableau

Entasser le tableau

Tri par tas

Activité 1 : Écrire la fonction `echanger(t: list, i1: int, i2: int) → None` qui inverse les éléments d'indice `i1` et `i2` du tableau `t`.

Avant de regarder la correction

Un tas

Définition

Tamiser un élément du tableau

Entasser le tableau

Tri par tas



- ▶ Prendre le temps de réfléchir,
- ▶ Analyser les messages d'erreur,
- ▶ Demander au professeur.

Un tas

Définition

Tamiser un élément du tableau

Entasser le tableau

Tri par tas

```
1 def echanger(t: list, i1: int, i2: int) -> None:
2     """
3     inverse les deux valeurs du tableau
4     """
5     t[i1], t[i2] = t[i2], t[i1]
```


1. Un tas

1.1 Définition

1.2 Tamiser un élément du tableau

1.3 Entasser le tableau

2. Tri par tas

Un tas

Définition

**Tamiser un élément du
tableau**

Entasser le tableau

Tri par tas

Tamiser un élément du tableau

Un tableau n'est pas nécessairement un tas (figure 2).

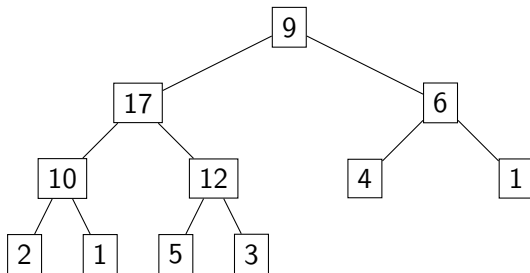


FIGURE 2 – La racine ne respecte pas les propriétés du tas

Un tas

Définition

Tamiser un élément du
tableau

Entasser le tableau

Tri par tas

Tamiser un élément du tableau consiste à faire respecter la propriété d'un arbre partiellement ordonné pour ce nœud. Prenons l'exemple du nœud racine contenant la valeur 9. Pour respecter la propriété il faut échanger la valeur du nœud père (9) avec la valeur de son plus grand fils (17).

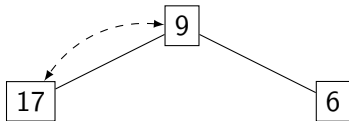


FIGURE 3 – Échange du père avec son plus grand fils.

Un tas

Définition

Tamiser un élément du tableau

Entasser le tableau

Tri par tas

Un tas

Définition

Tamiser un élément du
tableau

Entasser le tableau

Tri par tas

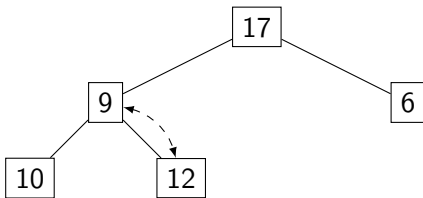


FIGURE 4 – Propagation récursive

Un tas

Définition

Tamiser un élément du
tableau

Entasser le tableau

Tri par tas

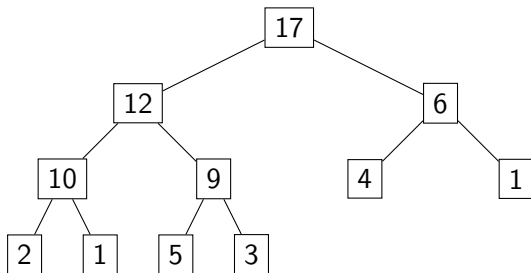


FIGURE 5 – La valeur 9 est à sa place

Un tas

Définition

Tamiser un élément du
tableau

Entasser le tableau

Tri par tas

Activité 2 : Écrire la fonction `get_i_fils_max(t: list, i_pere: int, i_max: int) → int` qui renvoie l'indice du plus grand fils du nœud d'indice `i_pere`. Si le père est plus grand que ses fils la fonction renverra `i_pere`. De plus la fonction n'ira pas jusqu'au bout du tableau mais s'arrêtera au nœud d'indice `i_max` (inclus). Les nœuds suivants seront considérés déjà triés et ne seront plus modifiés.

Avant de regarder la correction

Un tas

Définition

Tamiser un élément du tableau

Entasser le tableau

Tri par tas



- ▶ Prendre le temps de réfléchir,
- ▶ Analyser les messages d'erreur,
- ▶ Demander au professeur.

```
1 def get_i_fils_max(t: list, i_pere: int, i_max: int) -> int:
2     potentiel = i_pere
3     # regarde d'abord le gauche
4     i_gauche = 2*i_pere+1
5     if i_gauche <= i_max:
6         if t[i_gauche] > t[potentiel]:
7             potentiel = i_gauche
8
9     # puis le droit
10    i_droit = 2*i_pere+2
11    if i_droit <= i_max:
12        if t[i_droit] > t[potentiel]:
13            potentiel = i_droit
14
15    return potentiel
```


Un tas

Définition

Tamiser un élément du
tableau

Entasser le tableau

Tri par tas

Activité 3 : Écrire la fonction récursive `tamiser(t: list, i_pere: int, i_max: int) → None` qui respecte l'algorithme suivant :

- ▶ Trouver l'indice du nœud maximal entre `i_pere` et ses deux fils.
- ▶ Si l'indice n'est pas `i_pere` :
 - ▶ Échanger `i_pere` avec son fils le plus grand.
 - ▶ Tamiser ce nœud fils (qui vient de prendre la valeur de `i_pere`).

Avant de regarder la correction



- ▶ Prendre le temps de réfléchir,
- ▶ Analyser les messages d'erreur,
- ▶ Demander au professeur.

Un tas

Définition

Tamiser un élément du tableau

Entasser le tableau

Tri par tas

Un tas

Définition

Tamiser un élément du tableau

Entasser le tableau

Tri par tas

```
1 def tamiser(t: list, i_pere: int, i_max: int) -> None:
2     i_fils_max = get_i_fils_max(t, i_pere, i_max)
3     if i_fils_max != i_pere:
4         echanger(t, i_pere, i_fils_max)
5         tamiser(t, i_fils_max, i_max)
```

1. Un tas

1.1 Définition

1.2 Tamiser un élément du tableau

1.3 Entasser le tableau

2. Tri par tas

Un tas

Définition

Tamiser un élément du tableau

Entasser le tableau

Tri par tas

Un tas

Définition

Tamiser un élément du tableau

Entasser le tableau

Tri par tas

En tamisant chaque élément du tableau, on obtient un tas. Une bonne approche est de commencer par le bas de l'arbre. Ainsi, les sous-arbres droit et gauche respectent toujours la propriété d'un arbre partiellement ordonné. De plus, une feuille respectant nécessairement la propriété, il est judicieux de ne commencer qu'au premier nœud qui n'est pas une feuille.

Un tas

Définition

Tamiser un élément du
tableau

Entasser le tableau

Tri par tas

Activité 4 : Écrire la fonction `entasser(t: list) → None` qui transforme le tableau `t` en tas, en tamisant chaque nœud (qui n'est pas une feuille).

Avant de regarder la correction



- ▶ Prendre le temps de réfléchir,
- ▶ Analyser les messages d'erreur,
- ▶ Demander au professeur.

Un tas

Définition

Tamiser un élément du tableau

Entasser le tableau

Tri par tas

Un tas

Définition

Tamiser un élément du tableau

Entasser le tableau

Tri par tas

```
1 def entasser(t: list) -> None:
2     # indice du dernier noeud qui n'est pas
   une feuille
3     i = len(t)//2 - 1
4     while i >= 0:
5         tamiser(t, i, len(t)-1)
6         i -= 1
```


1. Un tas

2. Tri par tas

Un tas

Définition

Tamiser un élément du
tableau

Entasser le tableau

Tri par tas

Dans un tas la racine contient la plus grande valeur. Il suffit alors d'inverser cette valeur avec la dernière du tableau pour la placer définitivement. Il suffit ensuite d'appliquer le même principe au reste du tableau.

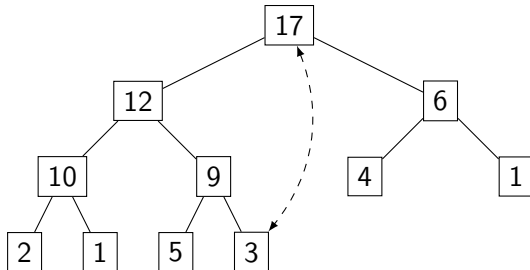


FIGURE 6 – Placer le plus grand élément en fin.

Un tas

Définition

Tamiser un élément du tableau

Entasser le tableau

Tri par tas

Un tas

Définition

Tamiser un élément du
tableau

Entasser le tableau

Tri par tas

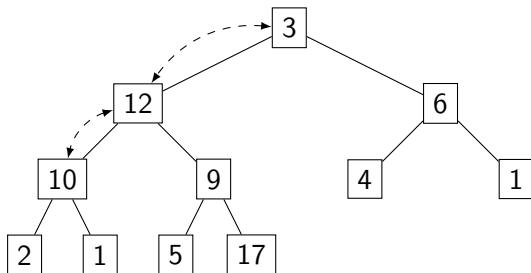


FIGURE 7 – Tamiser.

Un tas

Définition

Tamiser un élément du
tableau

Entasser le tableau

Tri par tas

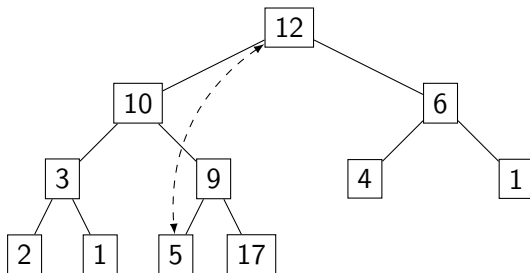


FIGURE 8 – Recommencer.

Un tas

Définition

Tamiser un élément du
tableau

Entasser le tableau

Tri par tas

Activité 5 : Écrire la fonction `tri_par_tas(t: list) → None` qui implémente l'algorithme suivant :

- ▶ Entasser le tableau.
- ▶ Initialiser `indice_dernier`.
- ▶ Tant que `indice_dernier > 0`
 - ▶ Échanger le premier élément avec l'élément d'indice `indice_dernier`.
 - ▶ Tamiser le tableau de la racine jusqu'à l'élément d'indice `indice_dernier`.
 - ▶ Décrémenter `indice_dernier`.

Avant de regarder la correction



- ▶ Prendre le temps de réfléchir,
- ▶ Analyser les messages d'erreur,
- ▶ Demander au professeur.

Un tas

Définition

Tamiser un élément du tableau

Entasser le tableau

Tri par tas

Un tas

Définition

Tamiser un élément du tableau

Entasser le tableau

Tri par tas

```
1 def tri_par_tas(t: list) -> None:
2     entasser(t)
3     dernier = len(t) - 1
4     while dernier > 0:
5         echanger(t, 0, dernier)
6         tamiser(t, 0, dernier-1)
7         dernier -= 1
```

Un tas

Définition

Tamiser un élément du
tableau

Entasser le tableau

Tri par tas

```
1 tab = [randint(0, 100) for _ in range(10)]
2 print(tab)
3 tri_par_tas(tab)
4 print(tab)
```

Code 2 – Tester la fonction

Un tas

Définition

Tamiser un élément du
tableau

Entasser le tableau

Tri par tas

Ce tri a une complexité temporelle en $O(n.\log(n))$