Informatique et Programmation

Appendice 3

Jean-Jacques Lévy

jean-jacques.levy@inria.fr

http://jeanjacqueslevy.net/prog-py-22

Plan

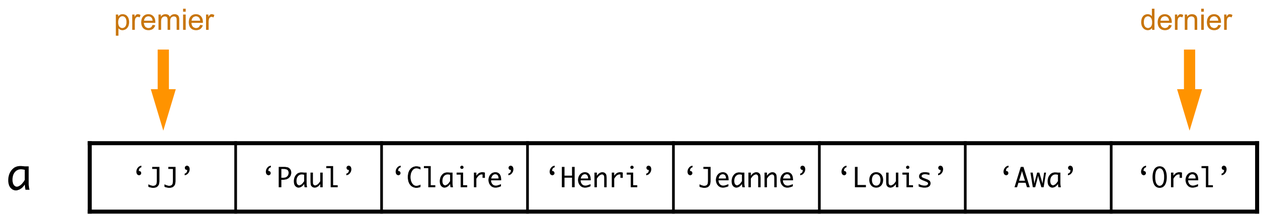
- classes et objets
- classes et héritage
- files d'attente

dès maintenant: télécharger Python 3 en `http://www.python.org`

un cours Python en `http://www.w3schools.com/python/default.asp`

Files d’attente

- une simple file d’attente (*FIFO — First In First Out*)




```
def ajouter_file (x, a) :
    a.append(x)


def enlever_file () :
    try:
      del a[0]
    except Exception:
      print ('erreur')
```

```
def nouvelle_file () :
    return [ ]

a = nouvelle_file ()
```

les sources sont en http://jeanjacqueslevy.net/prog-py-22/progs/c7a.py

Files d'attente

- une classe pour les FIFO


```
class FIFO :
    def __init__ (self, xs) :
        self.content = xs.copy()
        self.len = len(xs)

    def __str__ (self) :
        return '{}'.format (self.content)
```

```
f = FIFO(['JJ', 'Paul', 'Claire'])
print (f, f.len)
```

```
class FIFO :
    ...
    def push (self, x) :
        self.content.append (x)
        self.len += 1

    def pull (self) :
        if self.len > 0 :
            x = self.content[0]
            del (self.content[0])
            self.len -= 1
            return x
        else:
            print ('Erreur: file vide')
            raise Exception
```

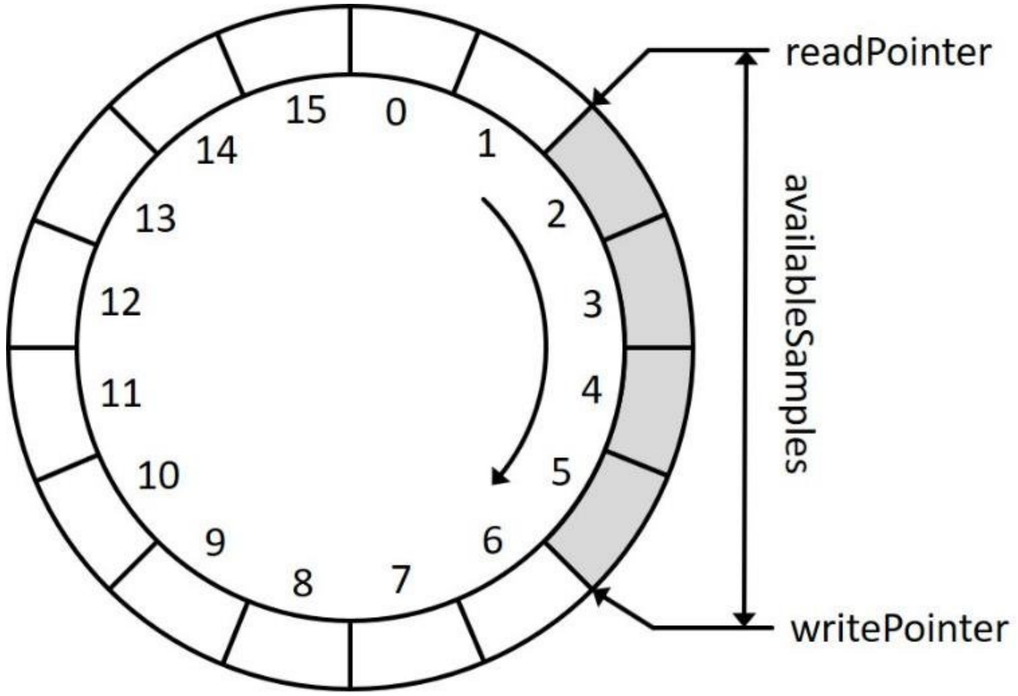
couteux ?

couteux ?

Exercice Ecrire l'addition de 2 files.

Files d’attente

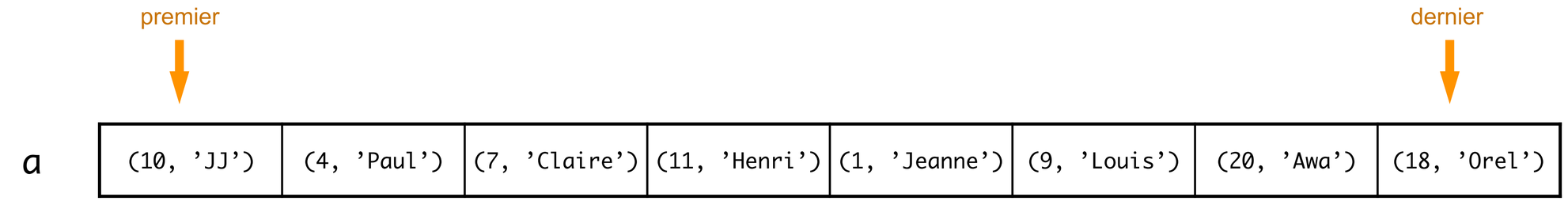
- implémentation avec un espace mémoire de taille fixe
- avec un tampon circulaire
- dans le hardware, il y a beaucoup de telles files d’attente



Exercice Ecrire un programme pour un tampon circulaire.

Files de priorité

• une file d'attente avec priorité [le plus prioritaire passe en premier]



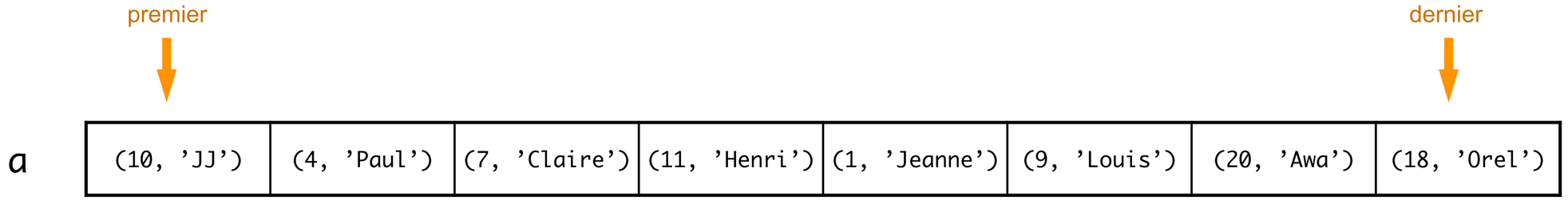

```
def ajouter_file (x, a) :
    a.append(x)


def enlever_file () :
    try:
      i = index_max_of(a)
      del a[i]
    except Exception:
      print ('erreur')
```

```
def nouvelle_file () :
    return [ ]

a = nouvelle_file ()
```

Exercice Modifier la classe précédente pour faire une classe des files de priorité.

Files de priorité

• une file d'attente avec priorité [le plus prioritaire passe en premier]



Exercice Modifier la classe précédente pour faire une classe des files de priorité. La solution est:

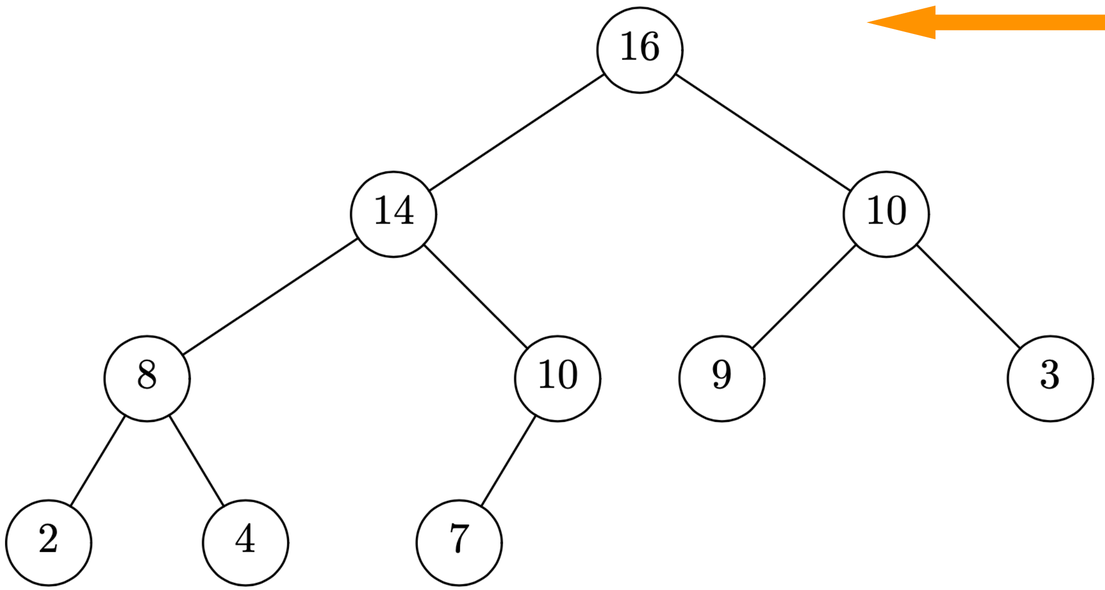
```
class PriorityQueue :
    ...

    def pull (self) :
            if self.len > 0 :
                i = index_of_max (self.content)
                x = self.content[i]
                del (self.content[i])
                self.len -= 1
                return x
            else:
                print ('Erreur: file vide')
                raise Exception
```

```
def index_of_max (a) :
    if len(a) > 0 :
        i = 0
        for j in range (1, len(a)) :
            if a[j] > a[i] :
                i = j
        return i
    else:
        raise Exception
```

```
f = PriorityQueue ([(4, 'Paul'), (10, 'JJ'), (5, 'Aline')])
print (f)
f.push ((5, 'Taka')); print (f)
x = f.pull(); print (x, f)
```

Files de priorité

• implémentation efficace [pour simplifier, on ne considère que les priorités]



maximum

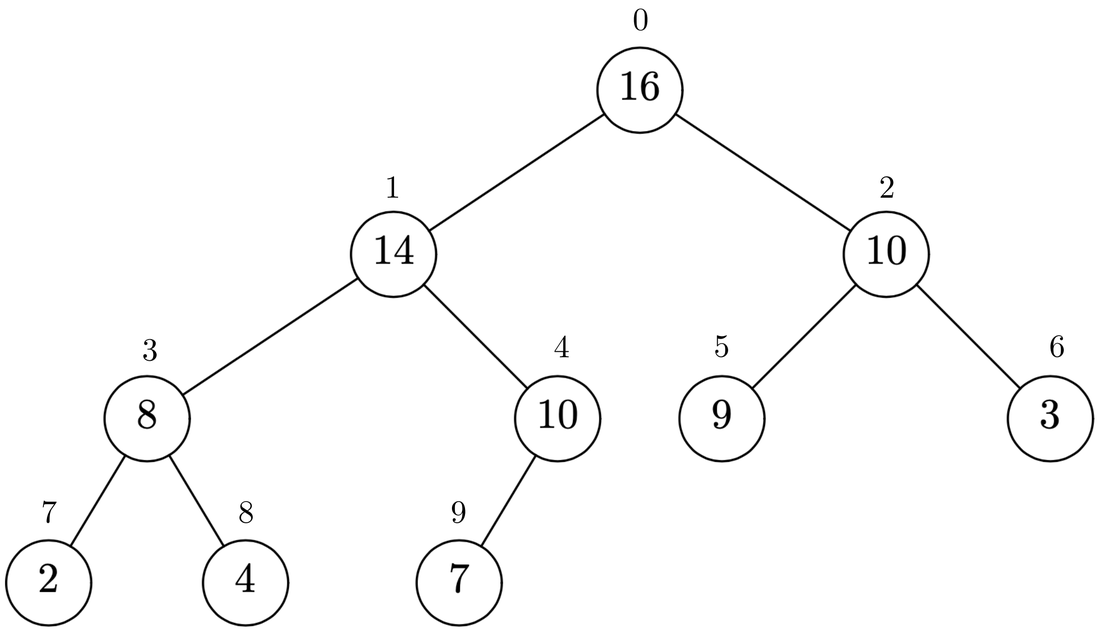
tous les noeuds internes ont 2 fils sauf les plus éloignés de la racine.

pour représenter une **file de priorité**, on utilise un arbre binaire presque parfait

[un ancêtre a une valeur plus élevée qu'un descendant]

Files de priorité

- on veut gérer une file d'attente où chacun a une priorité [le plus prioritaire passe en premier]



fils_gauche$(i) = 2i + 1$

fils_droit$(i) = 2i + 2$

père$(i) = \lfloor (i-1)/2 \rfloor$

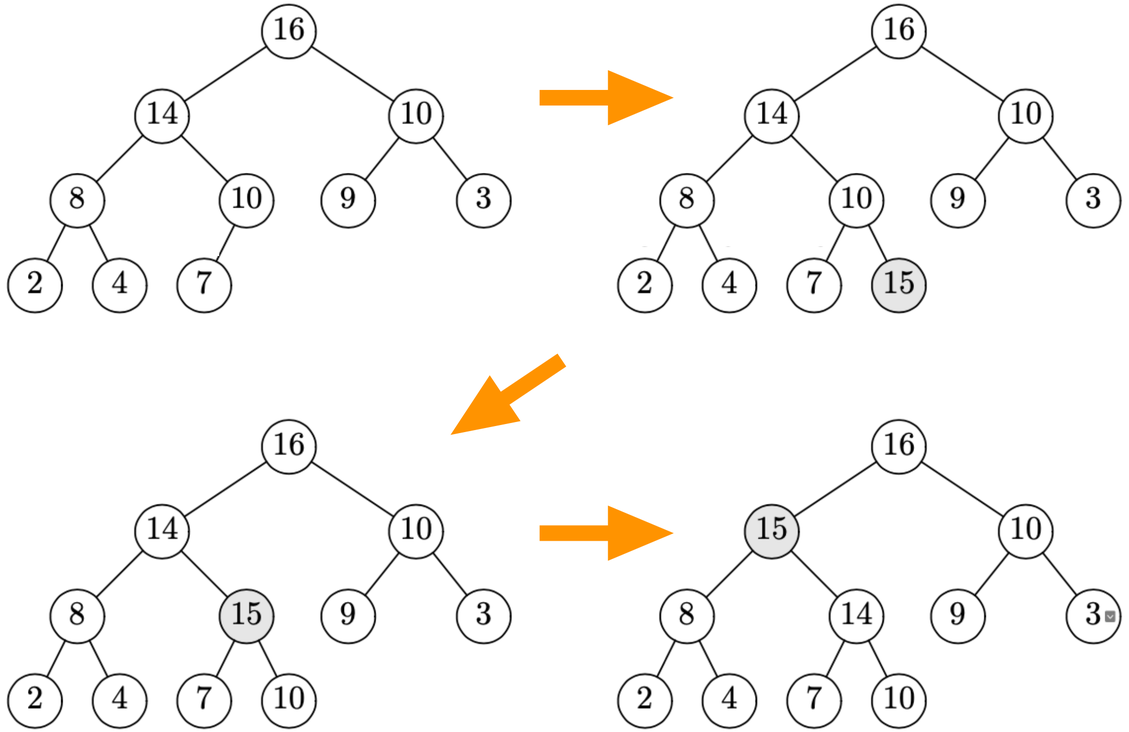
- on utilise un **tas** (*heap*) c'est-à-dire un tableau indicé par les numéros figurant au dessus de chaque noeud

i	0	1	2	3	4	5	6	7	8	9
$a[i]$	16	14	10	8	10	9	3	2	4	7

Files de priorité

• créer une file vide

```
def nouvelle_file () :
    return []

a = nouvelle_file ()
```



• ajouter un élément à la file

```
def ajouter_file (x, a) :
    a.append(x)
    n = len (a); i = n - 1
    while i > 0 and a[(i-1) // 2] < x :
        a[i] = a[(i-1) // 2]
        i = (i-1) // 2
    a[i] = x
```


• si n est la longueur de la file, l'ajout d'un élément ne fait pas plus que log(n) opérations

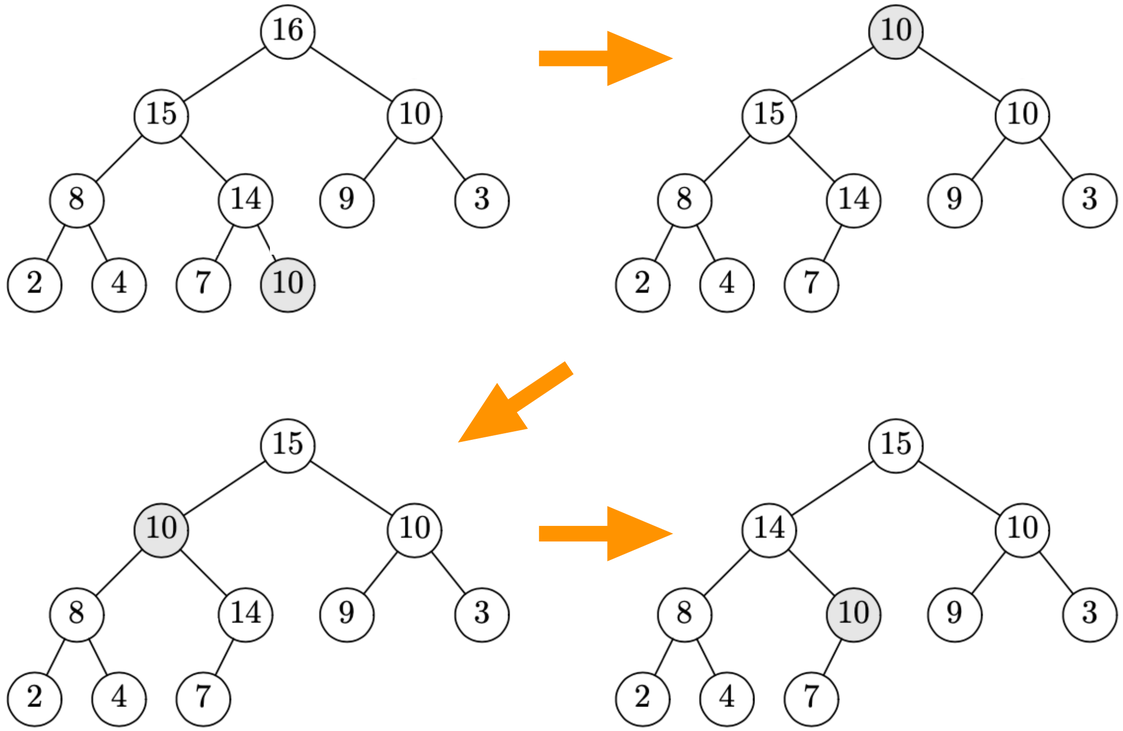
```
def max_de_file (a) :
    return a[0]
```

les sources sont en http://jeanjacqueslevy.net/prog-py-22/progs/c7a.py

Files de priorité

Exercice expliquer la fonction `enlever_file (a)` qui retire l'élément maximum de la file *a*

```
def enlever_file (a) :
    n = len (a)
    try:
        v = a[0] = a[n-1]
        del a[n-1]
        i = 0
        while 2*i + 1 < n-1 :
            j = 2*i + 1
            if j + 1 < n-1 :
                if a[j+1] > a[j] :
                    j = j + 1
            if v >= a[j] :
                break
            a[i] = a[j]; i = j
        a[i] = v
    except Exception:
        print ('erreur')
```



Exercice (*Heapsort*) écrire la fonction `trier_par_tas (a)` qui trie le tableau *a* en `nlog(n)` opérations à l'aide de tas

Files de priorité

Exercice Faire une classe de files de priorité avec une implantation par tas. La solution est :

```
class PriorityQueue :
    def __init__ (self, xs) :
        self.content = []
        self.len = 0
        for x in xs :
            self.push (x)

    def __str__ (self) :
        return '{}'.format (self.content)

    def push (self, x) :
        self.content.append(x)
        self.len += 1
        a = self.content; n = self.len
        i = n - 1
        while i > 0 and a[(i-1) // 2] < x :
            a[i] = a[(i-1) // 2]
            i = (i-1) // 2
        a[i] = x
```

couteux ?

```
class PriorityQueue :
    ...

    def pull (self) :
        if self.len > 0 :
            a = self.content; n = len (a);
            x = a[0]; v = a[0] = a[n-1]; del a[n-1]
            i = 0
            while 2*i + 1 < n-1 :
                j = 2*i + 1
                if j + 1 < n-1 :
                    if a[j+1] > a[j] :
                        j = j + 1
                if v >= a[j] :
                    break
                a[i] = a[j]; i = j
            a[i] = v
            return x
        else:
            print ('Erreur: file vide')
            raise Exception
```

couteux ?

Files de priorité

Exercice Faire une classe de files de priorité avec une implantation par tas dans un espace de taille fixe

Prochain cours

- les arbres en informatiques
- arbres de syntaxe abstraite
- parcours d’arbres
- arbres de recherche