

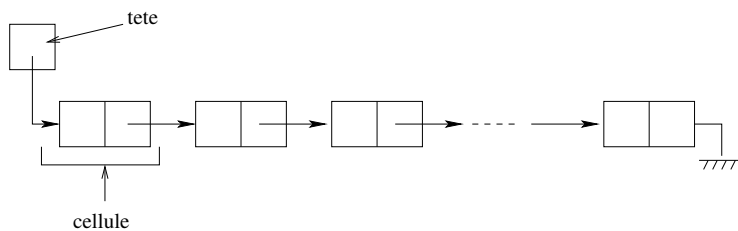
# Structures chaînées

## 1 Motivation

Aujourd'hui nous souhaitons manipuler des séquences d'éléments qui sont amenées à être modifiées fréquemment : soit en y insérant de nouveaux éléments, soit en supprimant des éléments qui s'y trouvent. Les tableaux représentent des séquences d'éléments mais sont inadaptés en cas de modification fréquente : une insertion ou une suppression en début ou milieu de tableau nous oblige à décaler tous les éléments qui suivent. Nous allons voir un autre type de structure de données, la liste chaînée, qui, au contraire, est performante pour ces opérations.

## 2 La liste chaînée

Le principe d'une liste chaînée est de représenter une séquence d'éléments sous la forme d'une liste de cellules (ou maillons) chaînées les unes aux autres. Chaque cellule contient un élément ainsi que la position de la cellule qui la suit dans la liste. L'ensemble de la liste est manipulé uniquement par la tête (première cellule) dont la position suffit pour retrouver tous les éléments. En Java, les cellules seront des objets et les positions des références à des cellules. La figure ci-dessous résume tout cela :



En Java :

```
class Cellule {  
    int elem;  
    Cellule suiv;  
}  
class Liste {  
    Cellule tete;  
    ...  
}
```

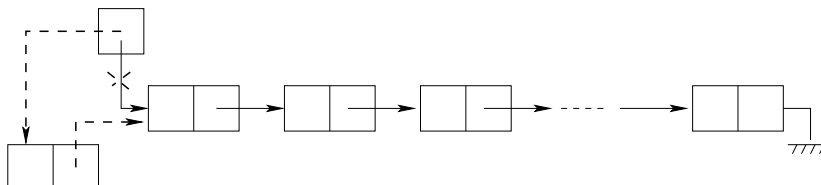
Les fichiers `Cellule.java` et `Liste.java` qui vous sont fournis contiennent une méthode d'affichage d'une liste chaînée et un petit exemple de construction cellule par cellule d'une liste à deux éléments.

## 3 Programmation des primitives de base de la liste chaînée

Votre mission est de programmer toutes les opérations permettant de manipuler une liste chaînée, ce qui se résume aux diverses opérations d'insertion ou de suppression suivantes :

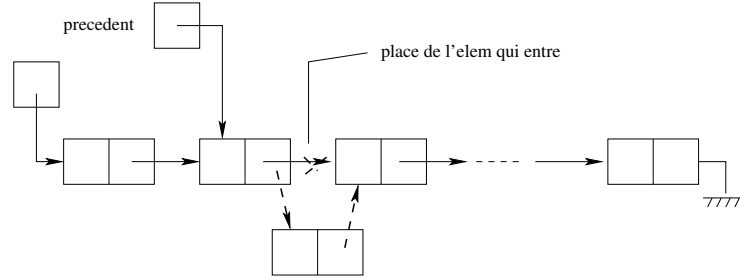
### 1. insérer en tête

Ici, l'idée est d'insérer un élément avant tous les autres. La nouvelle cellule associée devient alors la tête de liste.

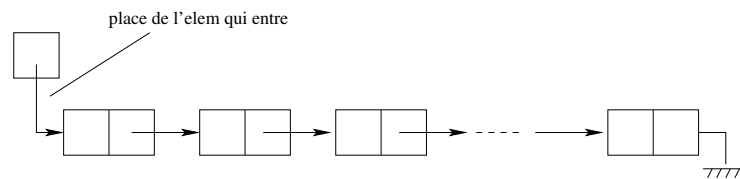


## 2. insérer à sa place

Lorsque la liste est triée, c'est-à-dire que les éléments sont placés dans la séquence par ordre croissant, il peut être intéressant d'insérer un nouvel élément à la bonne place afin de maintenir la liste triée. Dans ce cas, l'insertion début par une recherche de la bonne position avant de mettre à jour le chaînage.

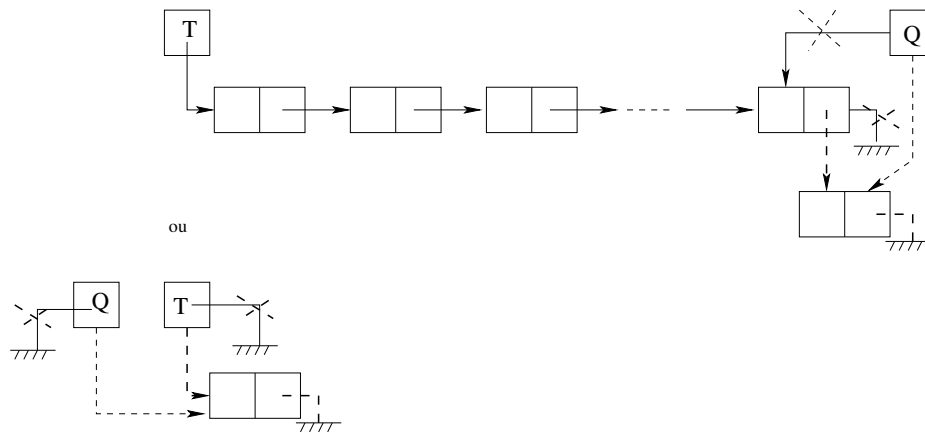


ou insertion en tete :



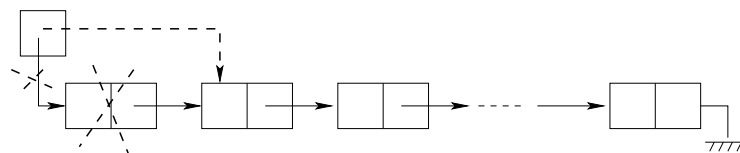
## 3. insérer en queue

Une solution naïve pour l'insertion en fin de liste ou insertion en queue, serait de parcourir toute la liste pour trouver la dernière cellule, puis d'insérer la nouvelle cellule à sa suite. Une meilleure solution serait de conserver dans la liste une référence à la dernière cellule, la queue. L'insertion revient alors à chaîner une nouvelle cellule après la queue et à mettre à jour la queue. Attention cependant, vous devez revenir sur les méthodes d'insertion précédentes qui doivent elles aussi maintenir une valeur correcte comme référence à la queue.



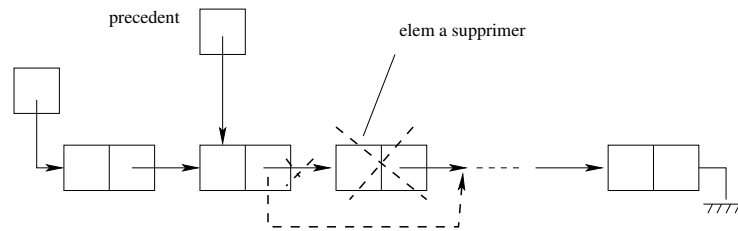
## 4. supprimer en tête

C'est le pendant de l'insertion en tête, il suffit juste de modifier la tête.

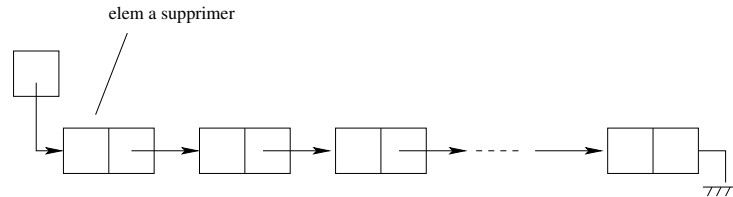


### 5. supprimer un élément donné

Ici, comme dans le cas de l'insertion à sa place, il faut parcourir la liste pour chercher l'élément car on ne peut pas deviner à l'avance où il se trouve.



ou suppression en tete :



Question subsidiaire : pourquoi ne vous demande-t-on pas d'écrire l'opération de suppression en queue ?

## 4 Structures de données pouvant être déduites de la liste chaînée

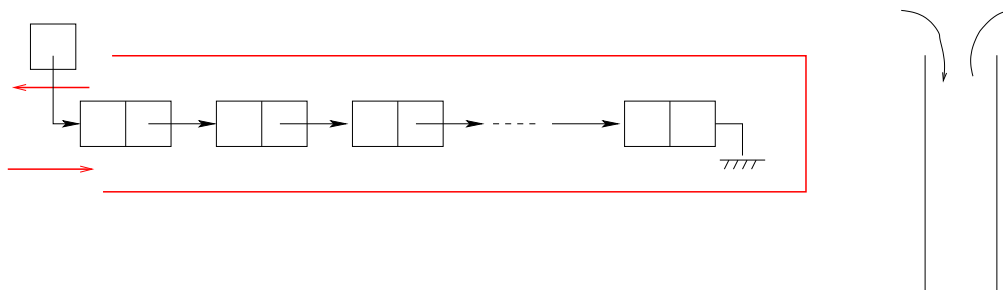
Ici, nous allons voir que plusieurs structures de données utiles peuvent se réaliser à l'aide d'une liste chaînée. La plupart des réalisations qu'on vous demande ici se résument à l'appel de la bonne méthode de la liste chaînée. Pensez bien à écrire un programme appelant vos méthodes sur des entiers choisis ou générés aléatoirement pour tester vos solutions.

### 4.1 Une pile

Avec une liste chaînée, il est possible de réaliser une structure de données appelée pile<sup>1</sup>. Une pile est une structure de données dotée de deux opérations principales :

- empiler, qui ajoute un nouvel élément au sommet de la pile ;
- dépiler<sup>2</sup>, qui supprime l'élément au sommet de la pile et renvoie sa valeur.

Une liste chaînée peut très bien servir à stocker une pile : on empile en insérant en tête et on dépile en supprimant en tête.



### 4.2 Une file (d'attente)

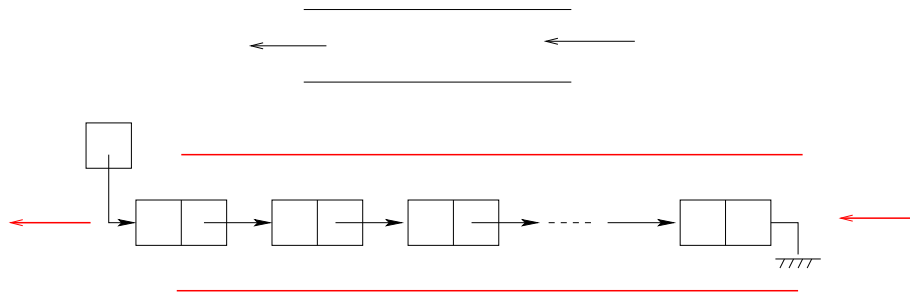
Avec une liste chaînée, il est possible de réaliser une structure de données appelée file<sup>3</sup>. Une file est une structure de données dotée de deux opérations principales :

- enfile, qui ajoute un nouvel élément en queue de file ;

1. il est aussi possible de réaliser une pile avec un tableau, mais ce n'est pas le propos ici  
2. oui, l'utilisation de ce nom dans ce contexte est une extension de la langue française  
3. ici aussi, il est possible de réaliser une file avec un tableau mais c'est plus compliqué à réaliser de manière efficace

— défiler<sup>4</sup>, qui supprime l'élément en tête de la file et renvoie sa valeur.

Une liste chaînée peut très bien servir à stocker une file : on enfila en insérant en queue et on défile en supprimant en tête.



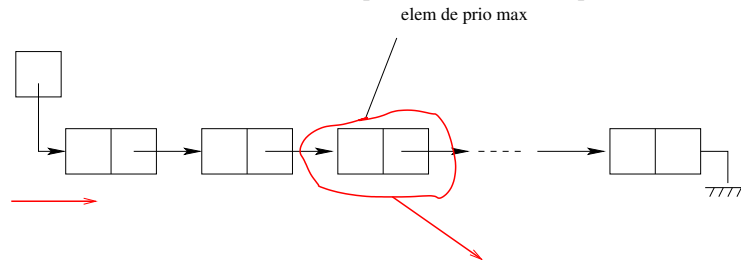
### 4.3 Une file d'attente à priorités

Avec une liste chaînée, il est possible de réaliser une structure de données appelée file d'attente à priorités ou FAP. Une FAP est une structure de données dotée de deux opérations principales :

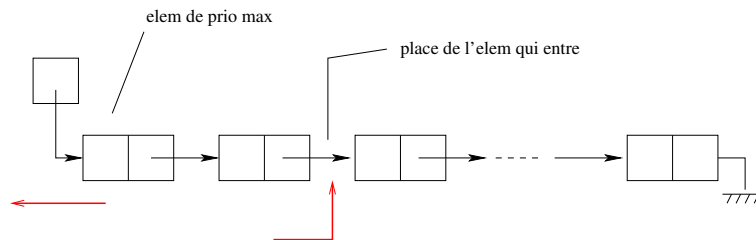
- insérer, qui étant donné un élément et une priorité, ajoute ce nouvel élément dans la FAP en lui attachant la priorité donnée ;
- défiler, qui supprime de la FAP l'élément de plus haute priorité et renvoie sa valeur.

Une liste chaînée peut très bien servir à stocker une FAP, il faut juste modifier les cellules pour stocker la valeur de l'élément ainsi que sa priorité. Dans un premier temps, pour simplifier le problème, nous supposerons que la valeur d'un élément est aussi sa priorité. Pour réaliser la FAP, nous avons deux stratégies possibles :

- insérer toujours en tête et rechercher l'élément de priorité maximale pour l'extraction ;



- insérer à sa place et supprimer l'élément en tête pour l'extraction. Attention, dans ce cas, la liste doit être triée par ordre décroissant<sup>5</sup>.



4. ici aussi, nous passerons sur les dérives que nous nous autorisons par rapport à la langue française

5. ou alors il faut inverser les valeurs