

Les listes

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Les listes

Implémentation
des listes

Réalisation
des opérations
primitives

Utilisation de
ces primitives

Les listes

Nour-Eddine Oussous, Éric Wegrzynowski

Licence ST-A, USTL - API2

26 octobre 2009

1 Introduction

2 Les listes

- Structure de liste
- Opérations primitives

3 Implémentation des listes

- Représentation contigüe
- Représentation chaînée

4 Réalisation des opérations primitives

- Constructeurs
- Sélecteurs
- Prédicat
- Opérations modificatrices

5 Utilisation de ces primitives

Les listes

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Les listes

Implémentation
des listes

Réalisation
des opérations
primitives

Utilisation de
ces primitives

Objectif :

Objectif :

- Trouver une représentation de la structure linéaire de liste

Objectif :

- Trouver une représentation de la structure linéaire de liste
- à l'aide de listes chaînées par des pointeurs

Objectif :

- Trouver une représentation de la structure linéaire de liste
- à l'aide de listes chaînées par des pointeurs
- et implémenter les opérations primitives à l'aide de cette représentation

Objectif :

- Trouver une représentation de la structure linéaire de liste
- à l'aide de listes chaînées par des pointeurs
- et implémenter les opérations primitives à l'aide de cette représentation
- de telle sorte que toute autre opération sur les listes puisse être implémentée à l'aide de ces opérations primitives sans manipulation explicite des pointeurs.

Notion de liste

Les listes

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Les listes

Implémentation des listes

Réalisation des opérations primitives

Utilisation de ces primitives

Définition (1)

liste = suite finie, éventuellement vide, d'éléments

- exemple : (3, 1, 4, 1, 5, 9, 2) est une liste d'entiers de longueur 7

Notion de liste

Les listes

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Les listes

Implémentation des listes

Réalisation des opérations primitives

Utilisation de ces primitives

Définition (1)

liste = suite finie, éventuellement vide, d'éléments

- exemple : (3, 1, 4, 1, 5, 9, 2) est une liste d'entiers de longueur 7
- une liste non vide possède

Notion de liste

Les listes

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Les listes

Implémentation des listes

Réalisation des opérations primitives

Utilisation de ces primitives

Définition (1)

liste = suite finie, éventuellement vide, d'éléments

- exemple : (3, 1, 4, 1, 5, 9, 2) est une liste d'entiers de longueur 7
- une liste non vide possède
 - 1 un élément de tête : 3

Notion de liste

Les listes

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Les listes

Implémentation des listes

Réalisation des opérations primitives

Utilisation de ces primitives

Définition (1)

liste = suite finie, éventuellement vide, d'éléments

- exemple : (3, 1, 4, 1, 5, 9, 2) est une liste d'entiers de longueur 7
- une liste non vide possède
 - 1 un élément de tête : 3
 - 2 et un reste qui est une liste : (1, 4, 1, 5, 9, 2)

Notion de liste

Les listes

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Les listes

Implémentation des listes

Réalisation des opérations primitives

Utilisation de ces primitives

Définition (1)

liste = suite finie, éventuellement vide, d'éléments

- exemple : $(3, 1, 4, 1, 5, 9, 2)$ est une liste d'entiers de longueur 7
- une liste non vide possède
 - 1 un élément de tête : 3
 - 2 et un reste qui est une liste : $(1, 4, 1, 5, 9, 2)$
- $()$ est la liste vide, de longueur 0

Notion de liste

Les listes

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Les listes

Implémentation des listes

Réalisation des opérations primitives

Utilisation de ces primitives

Définition (1)

liste = suite finie, éventuellement vide, d'éléments

- exemple : $(3, 1, 4, 1, 5, 9, 2)$ est une liste d'entiers de longueur 7
- une liste non vide possède
 - 1 un élément de tête : 3
 - 2 et un reste qui est une liste : $(1, 4, 1, 5, 9, 2)$
- $()$ est la liste vide, de longueur 0
- (2) est une liste de longueur 1 dont le reste est $()$

Structure récursive des listes

Les listes

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Les listes

Implémentation des listes

Réalisation des opérations primitives

Utilisation de ces primitives

Toute liste non vide peut être vue comme un couple
< tete; reste >.
Par exemple :

$$(3, 1, 4, 1, 5, 9, 2) = < \underbrace{3}_{\text{tete}} ; \underbrace{(1, 4, 1, 5, 9, 2)}_{\text{reste}} >$$

Structure réursive des listes

Les listes

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Les listes

Implémentation des listes

Réalisation des opérations primitives

Utilisation de ces primitives

Toute liste non vide peut être vue comme un couple
< tete; reste >.

Par exemple :

$$(3, 1, 4, 1, 5, 9, 2) = \underbrace{< 3 >}_{\text{tete}} ; \underbrace{(1, 4, 1, 5, 9, 2)}_{\text{reste}}$$

et en poursuivant sur les différents restes :

$$\begin{aligned} &= < 3; < 1; (4, 1, 5, 9, 2) >> \\ &\vdots \\ &= < 3; < 1; < 4; < 1; < 5; < 9; < 2; () >>>>>> \end{aligned}$$

Définition récursive

Les listes

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Les listes

Implémentation des listes

Réalisation des opérations primitives

Utilisation de ces primitives

Définition (2)

Une liste est donc

Définition récursive

Les listes

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Les listes

Implémentation des listes

Réalisation des opérations primitives

Utilisation de ces primitives

Définition (2)

Une liste est donc

1 soit la liste vide `()` ;

Définition récursive

Les listes

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Les listes

Implémentation des listes

Réalisation des opérations primitives

Utilisation de ces primitives

Définition (2)

Une liste est donc

- 1 soit la liste vide $()$;
- 2 soit un couple $\langle \text{tete}; \text{liste} \rangle$.

Définition récursive

Les listes

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Introduction

Les listes

Implémentation des listes

Réalisation des opérations primitives

Utilisation de ces primitives

Définition (2)

Une liste est donc

- 1 soit la liste vide $()$;
- 2 soit un couple $\langle \text{tete}; \text{liste} \rangle$.

En notant $Liste(E)$ l'ensemble des listes dont les éléments sont pris dans un ensemble E , on a

$$Liste(E) = \{()\} \cup (E \times Liste(E))$$

Ajout d'un élément en tête de liste

Les listes

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Les listes

Implémentation des listes

Réalisation des opérations primitives

Utilisation de ces primitives

Spécification

$$\begin{array}{lll} \text{ajouteEnTete} : & E \times \text{Liste}(E) & \longrightarrow \text{Liste}(E) \\ & e, l & \longmapsto \langle e; l \rangle \end{array}$$

Ajout d'un élément en tête de liste

Les listes

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Les listes

Implémentation
des listes

Réalisation
des opérations
primitives

Utilisation de
ces primitives

Spécification

$$\begin{array}{lll} \text{ajouteEnTete} : & E \times \text{Liste}(E) & \longrightarrow \text{Liste}(E) \\ & e, l & \longmapsto \langle e; l \rangle \end{array}$$

Exemple

$$\text{ajouteEnTete}(3, (1, 4, 1, 5, 9, 2)) = (3, 1, 4, 1, 5, 9, 2)$$

Ajout d'un élément en tête de liste

Les listes

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Les listes

Implémentation
des listes

Réalisation
des opérations
primitives

Utilisation de
ces primitives

Spécification

$$\begin{array}{lll} \text{ajouteEnTete} : & E \times \text{Liste}(E) & \longrightarrow \text{Liste}(E) \\ & e, l & \longmapsto \langle e; l \rangle \end{array}$$

Exemple

$$\text{ajouteEnTete}(3, (1, 4, 1, 5, 9, 2)) = (3, 1, 4, 1, 5, 9, 2)$$

C'est une opération de construction de liste.

Accès à la tête d'une liste

Les listes

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Les listes

Implémentation
des listes

Réalisation
des opérations
primitives

Utilisation de
ces primitives

Spécification

$$\begin{aligned} \text{tete} : \text{Liste}(E) &\longrightarrow E \\ < e; l > &\longmapsto e \end{aligned}$$

CU : La liste passée en paramètre ne doit pas être vide.

Accès à la tête d'une liste

Les listes

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Les listes

Implémentation des listes

Réalisation des opérations primitives

Utilisation de ces primitives

Spécification

$$\begin{aligned} \text{tete} : \text{Liste}(E) &\longrightarrow E \\ < e; l > &\longmapsto e \end{aligned}$$

CU : La liste passée en paramètre ne doit pas être vide.

Exemple

$$\text{tete}((3, 1, 4, 1, 5, 9, 2)) = 3$$

Accès à la tête d'une liste

Les listes

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Les listes

Implémentation des listes

Réalisation des opérations primitives

Utilisation de ces primitives

Spécification

$$\begin{aligned} \text{tete} : \text{Liste}(E) &\longrightarrow E \\ < e; l > &\longmapsto e \end{aligned}$$

CU : La liste passée en paramètre ne doit pas être vide.

Exemple

$$\text{tete}((3, 1, 4, 1, 5, 9, 2)) = 3$$

C'est un sélecteur.

Accès au reste d'une liste

Les listes

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Les listes

Implémentation des listes

Réalisation des opérations primitives

Utilisation de ces primitives

Spécification

$$\begin{aligned} \text{reste} : \text{Liste}(E) &\longrightarrow \text{Liste}(E) \\ < e; l > \longmapsto l \end{aligned}$$

CU : La liste passée en paramètre ne doit pas être vide.

Accès au reste d'une liste

Les listes

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Les listes

Implémentation des listes

Réalisation des opérations primitives

Utilisation de ces primitives

Spécification

$$\begin{aligned} \text{reste} : \text{Liste}(E) &\longrightarrow \text{Liste}(E) \\ \langle e; l \rangle &\longmapsto l \end{aligned}$$

CU : La liste passée en paramètre ne doit pas être vide.

Exemple

$$\text{reste}((3, 1, 4, 1, 5, 9, 2)) = (1, 4, 1, 5, 9, 2)$$

Accès au reste d'une liste

Les listes

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Les listes

Implémentation des listes

Réalisation des opérations primitives

Utilisation de ces primitives

Spécification

$$\begin{aligned} \text{reste} : \text{Liste}(E) &\longrightarrow \text{Liste}(E) \\ \langle e; l \rangle &\longmapsto l \end{aligned}$$

CU : La liste passée en paramètre ne doit pas être vide.

Exemple

$$\text{reste}((3, 1, 4, 1, 5, 9, 2)) = (1, 4, 1, 5, 9, 2)$$

C'est un sélecteur.

Test de vacuité d'une liste

Les listes

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Les listes

Implémentation des listes

Réalisation des opérations primitives

Utilisation de ces primitives

Spécification

$$\begin{array}{lll} \text{estListeVide} : \text{Liste}(E) & \longrightarrow & \text{Booleen} \\ & / & \longmapsto \begin{cases} \text{Vrai} & \text{si } l \text{ est vide} \\ \text{Faux} & \text{sinon} \end{cases} \end{array}$$

Test de vacuité d'une liste

Les listes

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Introduction

Les listes

Implémentation des listes

Réalisation des opérations primitives

Utilisation de ces primitives

Spécification

$$\begin{aligned} \text{estListeVide} : \text{Liste}(E) &\longrightarrow \text{Booleen} \\ l &\longmapsto \begin{cases} \text{Vrai} & \text{si } l \text{ est vide} \\ \text{Faux} & \text{sinon} \end{cases} \end{aligned}$$

Exemples

$$\begin{aligned} \text{estListeVide}() &= \text{Vrai} \\ \text{estListeVide}(3, 1, 4, 1, 5, 9, 2) &= \text{Faux} \end{aligned}$$

Test de vacuité d'une liste

Les listes

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Introduction

Les listes

Implémentation des listes

Réalisation des opérations primitives

Utilisation de ces primitives

Spécification

$$\begin{aligned} \text{estListeVide} : \text{Liste}(E) &\longrightarrow \text{Booleen} \\ l &\longmapsto \begin{cases} \text{Vrai} & \text{si } l \text{ est vide} \\ \text{Faux} & \text{sinon} \end{cases} \end{aligned}$$

Exemples

$$\begin{aligned} \text{estListeVide}() &= \text{Vrai} \\ \text{estListeVide}(3, 1, 4, 1, 5, 9, 2) &= \text{Faux} \end{aligned}$$

C'est un prédicat.

Changer la tête d'une liste

Les listes

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Les listes

Implémentation des listes

Réalisation des opérations primitives

Utilisation de ces primitives

Spécification

$$\begin{aligned} \text{modifierTete} : \text{Liste}(E), E &\longrightarrow \text{Liste}(E) \\ \langle e'; l \rangle, e &\longmapsto \langle e; l \rangle \end{aligned}$$

CU : La liste passée en paramètre ne doit pas être vide.

Changer la tête d'une liste

Les listes

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Les listes

Implémentation des listes

Réalisation des opérations primitives

Utilisation de ces primitives

Spécification

$$\begin{aligned} \text{modifierTete} : \text{Liste}(E), E &\longrightarrow \text{Liste}(E) \\ \langle e'; l \rangle, e &\longmapsto \langle e; l \rangle \end{aligned}$$

CU : La liste passée en paramètre ne doit pas être vide.

Exemple

$$\text{modifierTete}((5, 1, 4, 1, 5, 9, 2), 3) = (3, 1, 4, 1, 5, 9, 2)$$

Changer la tête d'une liste

Les listes

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Les listes

Implémentation des listes

Réalisation des opérations primitives

Utilisation de ces primitives

Spécification

$$\begin{aligned} \text{modifierTete} : \text{Liste}(E), E &\longrightarrow \text{Liste}(E) \\ \langle e'; l \rangle, e &\longmapsto \langle e; l \rangle \end{aligned}$$

CU : La liste passée en paramètre ne doit pas être vide.

Exemple

$$\text{modifierTete}((5, 1, 4, 1, 5, 9, 2), 3) = (3, 1, 4, 1, 5, 9, 2)$$

C'est une opération de modification de liste.

Changer le reste d'une liste

Les listes

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Les listes

Implémentation des listes

Réalisation des opérations primitives

Utilisation de ces primitives

Spécification

$$\begin{array}{lll} \text{modifierReste} : \text{Liste}(E), \text{Liste}(E) & \longrightarrow & \text{Liste}(E) \\ & & < e; l' >, l \longmapsto < e; l > \end{array}$$

CU : La première liste passée en paramètre ne doit pas être vide.

Changer le reste d'une liste

Les listes

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Les listes

Implémentation des listes

Réalisation des opérations primitives

Utilisation de ces primitives

Spécification

$$\begin{array}{lll} \text{modifierReste} : & \text{Liste}(E), \text{Liste}(E) & \longrightarrow \text{Liste}(E) \\ & \langle e; l' \rangle, l & \longmapsto \langle e; l \rangle \end{array}$$

CU : La première liste passée en paramètre ne doit pas être vide.

Exemple

$$\text{modifierReste}((3, 2), (1, 4, 1, 5, 9, 2)) = (3, 1, 4, 1, 5, 9, 2)$$

Changer le reste d'une liste

Les listes

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Les listes

Implémentation des listes

Réalisation des opérations primitives

Utilisation de ces primitives

Spécification

$$\begin{array}{lll} \text{modifierReste} : & \text{Liste}(E), \text{Liste}(E) & \longrightarrow \text{Liste}(E) \\ & \langle e; l' \rangle, l & \longmapsto \langle e; l \rangle \end{array}$$

CU : La première liste passée en paramètre ne doit pas être vide.

Exemple

$$\text{modifierReste}((3, 2), (1, 4, 1, 5, 9, 2)) = (3, 1, 4, 1, 5, 9, 2)$$

C'est une opération de modification de liste.

Représentation contigüe par tableaux

Les listes

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

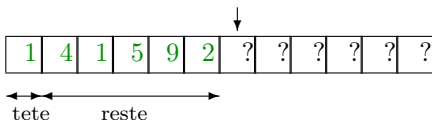
Introduction

Les listes

Implémentation des listes

Réalisation des opérations primitives

Utilisation de ces primitives



Représentation contigüe par tableaux

Les listes

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

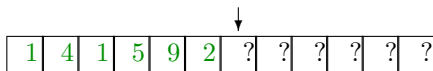
Introduction

Les listes

Implémentation des listes

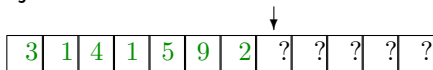
Réalisation des opérations primitives

Utilisation de ces primitives



tete reste

ajout en tête



tete reste

Inconvénients

Les listes

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Les listes

Implémentation
des listes

Réalisation
des opérations
primitives

Utilisation de
ces primitives

- réservation mémoire qui peut être inutilisée

Inconvénients

Les listes

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Les listes

Implémentation des listes

Réalisation des opérations primitives

Utilisation de ces primitives

- réservation mémoire qui peut être inutilisée
- ou insuffisante

Inconvénients

Les listes

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Les listes

Implémentation des listes

Réalisation des opérations primitives

Utilisation de ces primitives

- réservation mémoire qui peut être inutilisée
- ou insuffisante
- les opérations primitives sont coûteuses.

Objectifs

Les listes

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Les listes

**Implémentation
des listes**

Réalisation
des opérations
primitives

Utilisation de
ces primitives

Objectifs

Les listes

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Les listes

Implémentation
des listes

Réalisation
des opérations
primitives

Utilisation de
ces primitives

- occuper juste ce qu'il faut de mémoire

Objectifs

Les listes

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Les listes

Implémentation des listes

Réalisation des opérations primitives

Utilisation de ces primitives

- occuper juste ce qu'il faut de mémoire
- avoir des opérations primitives aussi peu coûteuses que possible

Moyen

Les listes

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Les listes

**Implémentation
des listes**

Réalisation
des opérations
primitives

Utilisation de
ces primitives

■ Utiliser l'allocation dynamique de mémoire

Moyen

Les listes

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Les listes

Implémentation des listes

Réalisation des opérations primitives

Utilisation de ces primitives

- Utiliser l'allocation dynamique de mémoire
- Une liste est un pointeur

Moyen

Les listes

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Les listes

Implémentation des listes

Réalisation des opérations primitives

Utilisation de ces primitives

- Utiliser l'allocation dynamique de mémoire
- Une liste est un pointeur
 - vers rien pour la liste vide (NIL)

Moyen

Les listes

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Les listes

Implémentation des listes

Réalisation des opérations primitives

Utilisation de ces primitives

- Utiliser l'allocation dynamique de mémoire
- Une liste est un pointeur
 - vers rien pour la liste vide (NIL)
 - vers un couple $\langle \text{tete}; \text{reste} \rangle$ que l'on nommera cellule

Les listes

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction


Les listes

Implémentation des listes

Réalisation des opérations primitives

Utilisation de ces primitives

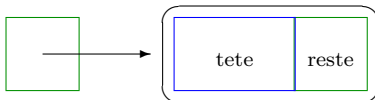
- La liste vide est un pointeur vers rien

$L =$ 

- La liste vide est un pointeur vers rien

$L =$ 

- Une liste non vide est un pointeur vers une cellule
< tete; reste >



Exemple de représentation chaînée

Les listes

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Introduction

Les listes

Implémentation
des listes

Réalisation
des opérations
primitives

Utilisation de
ces primitives

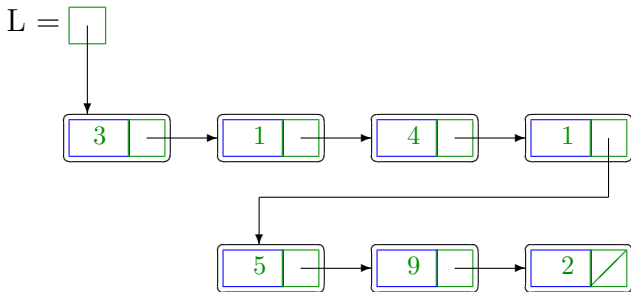


Fig.: Représentation chaînée de la liste $L = (3, 1, 4, 1, 5, 9, 2)$

Déclaration en Pascal

Les listes

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Les listes

Implémentation des listes

Réalisation des opérations primitives

Utilisation de ces primitives

```
LISTE      = ^CELLULE;  
CELLULE = record  
    info      : ELEMENT;  
    suivant   : LISTE;  
end {CELLULE};  
const  
    LISTEVIDE : LISTE = NIL;
```

Le type ELEMENT doit être déclaré par ailleurs.

Ajout en tête : algorithme

Les listes

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Introduction

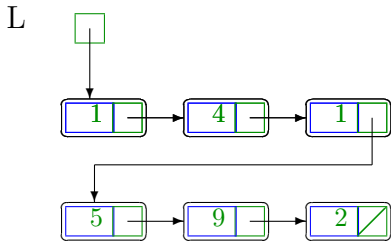
Les listes

Implémentation des listes

Réalisation des opérations primitives

Utilisation de ces primitives

Ajout de 3 en tête de la liste $L=(1,4,1,5,9,2)$.



Ajout en tête : algorithme

Les listes

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Introduction

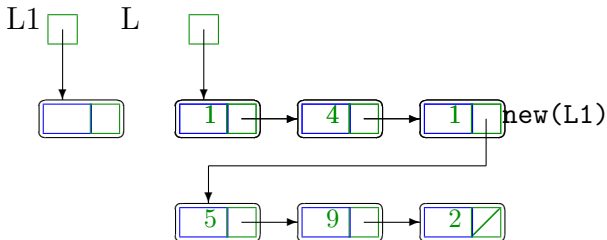
Les listes

Implémentation des listes

Réalisation des opérations primitives

Utilisation de ces primitives

Ajout de 3 en tête de la liste $L=(1,4,1,5,9,2)$.



Ajout en tête : algorithme

Les listes

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Introduction

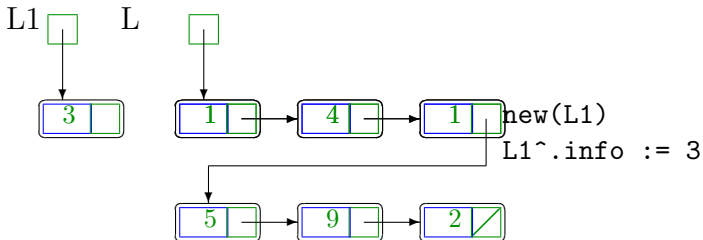
Les listes

Implémentation des listes

Réalisation des opérations primitives

Utilisation de ces primitives

Ajout de 3 en tête de la liste $L=(1,4,1,5,9,2)$.



Ajout en tête : algorithme

Les listes

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Introduction

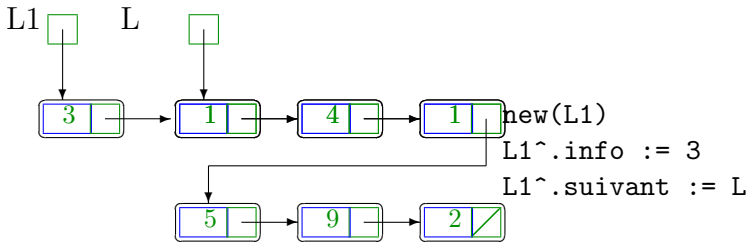
Les listes

Implémentation des listes

Réalisation des opérations primitives

Utilisation de ces primitives

Ajout de 3 en tête de la liste $L=(1,4,1,5,9,2)$.



Ajout en tête : algorithme

Les listes

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

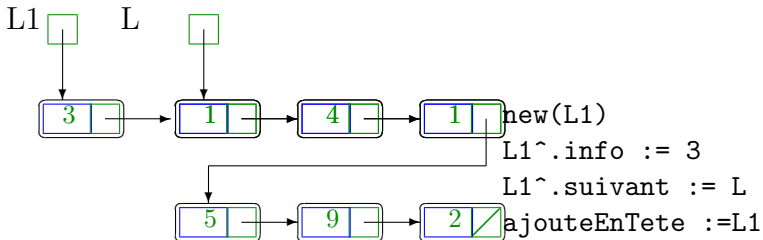
Les listes

Implémentation des listes

Réalisation des opérations primitives

Utilisation de ces primitives

Ajout de 3 en tête de la liste $L=(1,4,1,5,9,2)$.



Ajout en tête : algorithme

Les listes

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

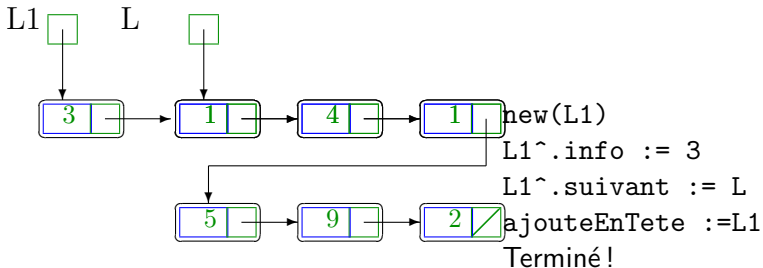
Les listes

Implémentation des listes

Réalisation des opérations primitives

Utilisation de ces primitives

Ajout de 3 en tête de la liste $L=(1,4,1,5,9,2)$.



La fonction ajouteEnTete

Les listes

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Les listes

Implémentation
des listes

Réalisation
des opérations
primitives

Utilisation de
ces primitives

```
// ajouteEnTete(e,l) = <e;l>
function ajouteEnTete(e : ELEMENT;
                      l : LISTE) : LISTE;

var
    l1 : LISTE;
begin
    new(l1);
    l1^.info := e;
    l1^.suivant := l;
    ajouteEnTete := l1;
end {ajouteEnTete};
```

Coût de l'ajout en tête

Les listes

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Les listes

Implémentation des listes

Réalisation des opérations primitives

Utilisation de ces primitives

- indépendant de la longueur de la liste

Coût de l'ajout en tête

Les listes

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Les listes

Implémentation des listes

Réalisation des opérations primitives

Utilisation de ces primitives

- indépendant de la longueur de la liste
- indépendant de la valeur des éléments de la liste

Coût de l'ajout en tête

Les listes

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Les listes

Implémentation des listes

Réalisation des opérations primitives

Utilisation de ces primitives

- indépendant de la longueur de la liste
- indépendant de la valeur des éléments de la liste
- indépendant de la valeur de l'élément ajouté

Coût de l'ajout en tête

Les listes

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Les listes

Implémentation des listes

Réalisation des opérations primitives

Utilisation de ces primitives

- indépendant de la longueur de la liste
- indépendant de la valeur des éléments de la liste
- indépendant de la valeur de l'élément ajouté

Conclusion

coût constant : $\Theta(1)$

La fonction tete

Les listes

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Les listes

Implémentation des listes

Réalisation des opérations primitives

Utilisation de ces primitives

```
//tete(l)=le premier element de la liste l
//CU : l non vide
function tete(l : LISTE) : ELEMENT;
begin
    tete := l^.info;
end {tete};
```

Coût de l'accès à la tête

Les listes

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Les listes

Implémentation des listes

Réalisation des opérations primitives

Utilisation de ces primitives

- indépendant de la longueur de la liste

Coût de l'accès à la tête

Les listes

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Les listes

Implémentation des listes

Réalisation des opérations primitives

Utilisation de ces primitives

- indépendant de la longueur de la liste
- indépendant de la valeur des éléments de la liste

Coût de l'accès à la tête

Les listes

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Les listes

Implémentation des listes

Réalisation des opérations primitives

Utilisation de ces primitives

- indépendant de la longueur de la liste
- indépendant de la valeur des éléments de la liste

Conclusion

coût constant : $\Theta(1)$

La fonction `reste`

Les listes

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Les listes

Implémentation des listes

Réalisation des opérations primitives

Utilisation de ces primitives

```
// reste(l) = la liste qui suit le premier
//           element de l
// CU : l non vide
function reste(l : LISTE) : LISTE;
begin
    reste := l^.suivant;
end {reste};
```

Coût de l'accès au reste

Les listes

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Les listes

Implémentation des listes

Réalisation des opérations primitives

Utilisation de ces primitives

- indépendant de la longueur de la liste

Coût de l'accès au reste

Les listes

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Les listes

Implémentation des listes

Réalisation des opérations primitives

Utilisation de ces primitives

- indépendant de la longueur de la liste
- indépendant de la valeur des éléments de la liste

Coût de l'accès au reste

Les listes

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Les listes

Implémentation des listes

Réalisation des opérations primitives

Utilisation de ces primitives

- indépendant de la longueur de la liste
- indépendant de la valeur des éléments de la liste

Conclusion

coût constant : $\Theta(1)$

Le prédicat `estListeVide`

Les listes

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Les listes

Implémentation
des listes

Réalisation
des opérations
primitives

Utilisation de
ces primitives

```
// estListeVide(l)  $\Leftrightarrow$  l est vide
function estListeVide(l: LISTE): BOOLEAN;
begin
    estListeVide := l = NIL;
end {estListeVide};
```

Coût du test de vacuité

Les listes

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Les listes

Implémentation des listes

Réalisation des opérations primitives

Utilisation de ces primitives

- indépendant de la longueur de la liste

Coût du test de vacuité

Les listes

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Les listes

Implémentation des listes

Réalisation des opérations primitives

Utilisation de ces primitives

- indépendant de la longueur de la liste
- indépendant de la valeur des éléments de la liste

Coût du test de vacuité

Les listes

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Les listes

Implémentation des listes

Réalisation des opérations primitives

Utilisation de ces primitives

- indépendant de la longueur de la liste
- indépendant de la valeur des éléments de la liste

Conclusion

coût constant : $\Theta(1)$

La procédure `modifierTete`

Les listes

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Les listes

Implémentation
des listes

Réalisation
des opérations
primitives

Utilisation de
ces primitives

```
// modifierTete(l,e) modifie la valeur de
// l'element en tete de l
// CU : l non vide
procedure modifierTete(const l : LISTE;
                        const e : ELEMENT);

begin
    l^.info := e ;
end {modifierTete};
```

La procédure `modifierTete`

Les listes

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Les listes

Implémentation
des listes

Réalisation
des opérations
primitives

Utilisation de
ces primitives

```
// modifierTete(l,e) modifie la valeur de
// l'element en tete de l
// CU : l non vide
procedure modifierTete(const l : LISTE;
                        const e : ELEMENT);

begin
    l^.info := e ;
end {modifierTete};
```

Remarque : le paramètre `l` est constant car c'est un pointeur.

Coût de la modification de la tête

Les listes

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Les listes

Implémentation des listes

Réalisation des opérations primitives

Utilisation de ces primitives

- indépendant de la longueur de la liste

Coût de la modification de la tête

Les listes

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Les listes

Implémentation des listes

Réalisation des opérations primitives

Utilisation de ces primitives

- indépendant de la longueur de la liste
- indépendant de la valeur des éléments de la liste

Coût de la modification de la tête

Les listes

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Les listes

Implémentation des listes

Réalisation des opérations primitives

Utilisation de ces primitives

- indépendant de la longueur de la liste
- indépendant de la valeur des éléments de la liste
- indépendant de la nouvelle valeur de la tête

Coût de la modification de la tête

Les listes

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Les listes

Implémentation des listes

Réalisation des opérations primitives

Utilisation de ces primitives

- indépendant de la longueur de la liste
- indépendant de la valeur des éléments de la liste
- indépendant de la nouvelle valeur de la tête

Conclusion

coût constant : $\Theta(1)$

La procédure `modifierReste`

Les listes

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Introduction

Les listes

Implémentation
des listes

Réalisation
des opérations
primitives

Utilisation de
ces primitives

```
// modifierReste(l,ll) modifie la valeur
// du reste de l en lui attribuant ll
// CU : l non vide
procedure modifierReste(const l : LISTE;
                        const ll : LISTE);

begin
    l^.suivant := ll;
end {modifierReste};
```

La procédure `modifierReste`

Les listes

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Introduction

Les listes

Implémentation
des listes

Réalisation
des opérations
primitives

Utilisation de
ces primitives

```
// modifierReste(l,ll) modifie la valeur
// du reste de l en lui attribuant ll
// CU : l non vide
procedure modifierReste(const l : LISTE;
                        const ll : LISTE);

begin
    l^.suivant := ll;
end {modifierReste};
```

Remarque : le paramètre `l` est constant car c'est un pointeur.

Coût de la modification du reste

Les listes

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Les listes

Implémentation des listes

Réalisation des opérations primitives

Utilisation de ces primitives

- indépendant de la longueur de la liste

Coût de la modification du reste

Les listes

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Les listes

Implémentation des listes

Réalisation des opérations primitives

Utilisation de ces primitives

- indépendant de la longueur de la liste
- indépendant de la valeur des éléments de la liste à modifier

Coût de la modification du reste

Les listes

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Les listes

Implémentation des listes

Réalisation des opérations primitives

Utilisation de ces primitives

- indépendant de la longueur de la liste
- indépendant de la valeur des éléments de la liste à modifier
- indépendant de la valeur des éléments du nouveau reste.

Coût de la modification du reste

Les listes

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Les listes

Implémentation des listes

Réalisation des opérations primitives

Utilisation de ces primitives

- indépendant de la longueur de la liste
- indépendant de la valeur des éléments de la liste à modifier
- indépendant de la valeur des éléments du nouveau reste.

Conclusion

coût constant : $\Theta(1)$

Utilisation des primitives

Les listes

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Les listes

Implémentation des listes

Réalisation des opérations primitives

Utilisation de ces primitives

```
{ L une liste qqe, e est un element qqe}  
L1 := ajouteEnTete(e, L) ;  
{ L1 = <e; L> }
```

Utilisation des primitives

Les listes

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Les listes

Implémentation des listes

Réalisation des opérations primitives

Utilisation de ces primitives

```
{ L une liste qqe, e est un element qqe}  
L1 := ajouteEnTete(e, L) ;  
{ L1 = <e; L> }
```

```
{ L = <t; R> une liste non vide}  
e := tete(L) ;  
{ e = t et L est inchangée }
```

Utilisation des primitives

Les listes

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Les listes

Implémentation des listes

Réalisation des opérations primitives

Utilisation de ces primitives

```
{ L une liste qqe, e est un element qqe}  
L1 := ajouteEnTete(e, L) ;  
{ L1 = <e; L> }
```

```
{ L = <t; R> une liste non vide}  
e := tete(L) ;  
{ e = t et L est inchangée }
```

```
{ L = <t; R> une liste non vide}  
L1 := reste(L) ;  
{ L1 = R et L est inchangée }
```

Utilisation des primitives

Les listes

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Les listes

Implémentation des listes

Réalisation des opérations primitives

Utilisation de ces primitives

```
{ L = <t; R> une liste non vide,  
      e est un element qqe}  
modifierTete(L, e) ;  
{ L = <e; R> et L est inchangee }
```

Utilisation des primitives

Les listes

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Les listes

Implémentation des listes

Réalisation des opérations primitives

Utilisation de ces primitives

```
{ L = <t; R> une liste non vide,  
      e est un element qqe}  
modifierTete(L, e) ;  
{ L = <e; R> et L est inchangee }
```

```
{ L = <t; R> une liste non vide,  
      S est une liste qqe}  
modifierReste(L, S) ;  
{ L = <t; S> et L est inchangee }
```