Représentation des données types construits

Christophe Viroulaud

Formation NSI

NSI

Représentation des données types construits

Les programmes

En première

Lii teiiiiii

Listes

Ambiguït

Tableau

Tubicuu

Complexité temporelle

Listo shaînéo

Définition

21 - D ---

118t Fython

et dictionnaires

Difficultés rencontrées

Surmonter les difficult

onction et aramètre mutal

Piles, files

nies, mes

Interface

Identifier et définir les différentes structures de données devant être maîtrisées en NSI.

Représentation des données types construits

Les programmes

En premièr

Listes

Ambiguïté

Tableau

C----Inition

Complexité temporelle

D. C. . .

Définition

list Pytho

Parcours de listes

Difficultés rencontrées

onction et

iles files

Piles, files

Interface

Représentation des données types construits

1. Les programmes

- 1.1 En première
- 1.2 En terminale
- 2 Listes
- 3. Parcours de listes et dictionnaires
- 4. Fonction et paramètre mutable
- 5. Piles. files

Les programmes

En première

Lictor

_13103

- III

Tableau

Difficulties

Complexité temporelle

iste chaînée

Définition

Complexité 1

list Python

Parcours de listes

et dictionnaires

Difficultés rencontrées

Surmonter les difficultés

aramètre m

Piles files

riles, files

Définition

Interface

Les programmes - En première

Tableau indexé, tableau donné en compréhension	Lire et modifier les éléments d'un tableau grâce à leurs index. Construire un tableau par compréhension. Utiliser des tableaux de tableaux pour représenter des matrices : notation a [i] [j]. Itérer sur les éléments d'un tableau.	Seuls les tableaux dont les éléments sont du même type son présentés. Aucune connaissance des tranches (slices) n'est exigible. L'aspect dynamique des tableaux de Python n'est pas évoqué. Python identifie listes et tableaux. Il n'est pas fait référence aux tableaux de la bibliothèque NumPy.		
Dictionnaires par clés et valeurs	Construire une entrée de dictionnaire. Itérer sur les éléments d'un dictionnaire.	Il est possible de présenter les données EXIF d'une image sous la forme d'un enregistrement. En Python, les p-uplets nommés sont implémentés par des dictionnaires. Utiliser les méthodes keys(), values () et items ().		

Représentation des données types construits

os programmos

En première

n torminale

tes

Ambiguïté

inbiguite

ableau

Définition

Complexité temporelle

DVC 11

Définition

Complexité t

list Python

arcours de

et dictionnaires

Surmonter les difficu

Surmonter les diffi

aramètre mutal

iles, files

Définition

nterface mplémentatio

- 1. Les programmes
- 1.2 En terminale

Représentation des données types construits

En terminale

En terminale

Structures de données, interface et implémentation.	données par son interface. Distinguer interface et implémentation. Écrire plusieurs implémentations	L'abstraction des structures de données est introduite après plusieurs implémentations d'une structure simple comme la file (avec un tableau ou avec deux piles).
---	---	--

Listes, piles, files : structures linéaires. Dictionnaires, index et clé.		On distingue les modes FIFO (first in first out) et LIFO (last in first		
		out) des piles et des files.		
	Choisir une structure de données adaptée à la situation à modéliser.			
	Distinguer la recherche d'une valeur dans une liste et dans un dictionnaire.			

Représentation des données types construits

Les programmes

En terminale

stes

Ambiguite

Tableau

Complexité temporelle

Définition

Complexité terr

list Python

et dictionnaires

Difficultés rencontrées

Surmonter les difficultés

aramètre mutab

Piles, files

Définition

2. Listes

- 2.4 list Python

Représentation des données types construits

Listes - Ambiguïté

Observation

Les list Python ne sont pas des listes.

Représentation des données types construits

Les programmes

En première

Lietae

Ambiguïté

. . . .

Définition

Complexité temporelle

Liste Chamee

Définition

11 of Python

Parcours de listes

Parcours de listes et dictionnaires

Difficultés rencontrées urmonter les difficultés

onction et Iramètre mutab

iles, files

Piles, files

Interface

Définition

Une liste est une séquence ordonnée d'éléments de même type. Chaque élément est repéré par sa position dans la liste.

Représentation des données types construits

Ambiguïté

Représentation des données types construits

2. Listes

2.2 Tableau Définition

Complexité temporelle

2.4 list Python

Tableau

Tableau - Définition

Définition

Un tableau est une séquence ordonnée et <u>contigüe</u> d'éléments de même type.

Le contenu d'un tableau est modifiable.

Représentation des données types construits

Les programmes

En première

Listes

Ambiguïté Tableau

Définition

Complexité temporel

Complexite temporelle

Définition

Complexité temporelle

Parcours de listes

Difficultés rencontrées Surmonter les difficultés

onction et aramètre mutab

iles, files

Définition Interface

h	е	I	I	0	ļ					
	3						9			
								6		
h	е	у	8	5	3	9	1	0	2	!
	3	4								

FIGURE 1 – Simulation de représentation en mémoire

Représentation des données types construits

Définition

Complexité temporelle

0	1	2	3	4	5	6
8	5	3	9	1	0	2

FIGURE 2 – On accède à un élément en temps constant grâce à son indice.

Représentation des données types construits

Complexité temporelle

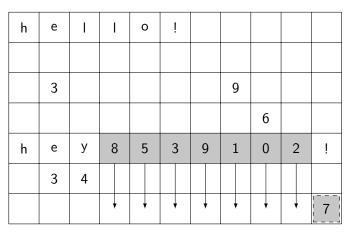


FIGURE 3 – Agrandir un tableau peut prendre **un temps linéaire** à la taille du tableau.

Représentation des données types construits

Les programmes
En première

Listes

Ambiguïté Tableau

Définition

Complexité temporelle

Complexite temporelle

Définition

Complexité to

Parcours de listes et dictionnaires

Difficultés rencontrées

onction et

ramètre mutabl

Piles, files
Définition

Interface

Représentation des données types construits

1. Les programmes

2. Listes

2.1 Ambiguïté

2.2 Tableau

2.3 Liste chaînée
Définition
Complexité temporelle

2.4 list Python

3. Parcours de listes et dictionnaires

4. Fonction et paramètre mutable

5. Piles, files

Les programmes

En première

Links

istes

mbiguïté

ableau

Définition

Complexité te

Complexite te

Liste chaînée

Définition

Jetinition

Complexité te

list Python

arcours de listes

t dictionnaires

Surmonter les difficultés

Fonction et paramètre mi

les, files

riies, iiies

Interface

Interface

Liste chaînée - Définition

Définition

Une liste chaînée est une séquence ordonnée où chaque élément possède une référence vers le suivant.

Représentation des données types construits

Les programmes

En première

listes

Ambiguïté

Tableau

Complexité temporelle

Complexite temporelle

Définition

Complexité tempore

list Python

Parcours de listes et dictionnaires

Difficultés rencontrées Surmonter les difficultés

onction et

iles files

Piles, files

Interface

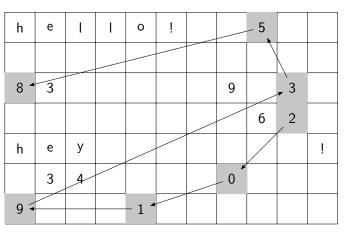


FIGURE 4 – Simulation de représentation en mémoire

Représentation des données types construits

Définition

Complexité temporelle

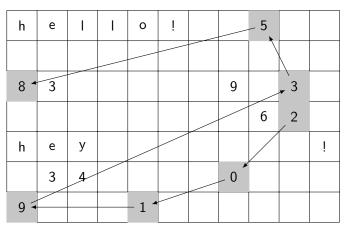


FIGURE 5 – On accède à l'élément de rang ${\tt n}$ en temps linéaire.

Représentation des données types construits

Les programmes

En première

istes

Ambiguïté

Tableau

Complexité temporelle

Ste chainee Définition

Complexité temporelle list Python

Parcours de listes

et dictionnaires Difficultés rencontrées

Surmonter les difficulté

aramètre m

Piles, files

Définition Interface

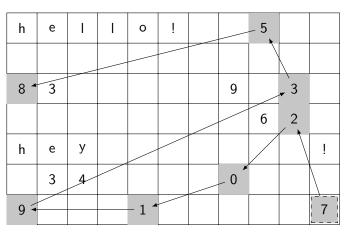


FIGURE 6 – On ajoute un élément en tête de liste **en temps constant**.

Représentation des données types construits

Les programmes

n premiere

istes

Ambiguïté Tableau

Tableau

Complexité temporelle

Liste chaînée

Définition

Complexité temporelle list Python

Parcours de listes et dictionnaires

Difficultés rencontrées Surmonter les difficulté

onction et aramètre mutal

Piles, files

Définition Interface

- 2. Listes

- 2.4 list Python

Représentation des données types construits

list Python

list Python

Observations

- Python est un langage de haut-niveau.
- Les list essaie de prendre les avantages des deux structures précédentes.
- Les list vont plus loin : les éléments peuvent être de types différents.

Représentation des données types construits

Les programmes

En première

istes

Ambiguïté

Гаbleau

Définition

Complexité temporelle

Liste chainee

Définition

Complexité temp

list Python

Parcours de listes

Difficultés rencontrées

Surmonter les di

onction et aramètre mutabl

Piles, files

Définition

Représentation des données types construits

1. Les programmes

2 Lister

- 3. Parcours de listes et dictionnaires
- 3.1 Difficultés rencontrées
- 3.2 Surmonter les difficultés
- 4. Fonction et paramètre mutable
- 5. Piles, files

Les programmes

En première

Listes

.13103

T-LI---

Tableau

Définition

Complexité temporelle

iste chaînée

Définition

Complexité:

list Python

list Python

Parcours de listes

Difficultés rencontrées

Surmonter les difficulté

aramètre m

iles files

riies, iiies

Interface

Parcours de listes et dictionnaires - Difficultés rencontrées

Constat

- Dans un tableau, confusion indice / élément.
- ► Tentative de récupération des informations dans un dictionnaire, avec un indice.

Représentation des données types construits

Les programmes

En première

listes

Ambiguïté

Tableau

Complexité temporelle

Liste chaînée

Définition

11st Python

Parcours de listes

et dictionnaires

Difficultés rencontrées

Surmonter les difficultés

onction et Iramètre mutab

Piles, files

Définition

```
# confusion indice / élément
tab = [3, 9, 1]
for i in tab:
    print(tab[i])
```

```
# indice dans un dictionnaire
dico = {"a": 2, "e": 4, "f": 1}
for i in range(len(dico)):
    print(dico[i])
```

Code 1 – Erreurs courantes

Représentation des données types construits

_es programmes

En première

.

istes

r i i

Fableau

etinition

Complexité temporelle

iste chainee

Delinition

Complexite tempore

et dictionnaires

Difficultés rencontrées

Surmonter les difficultés

arametre mut

Piles, files

Définition Interface

- 3. Parcours de listes et dictionnaires
- 3.2 Surmonter les difficultés

Représentation des données types construits

Surmonter les difficultés

Surmonter les difficultés

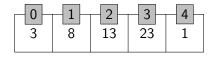


FIGURE 7 – Dans un tableau chaque élément possède une étiquette.

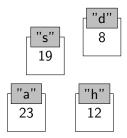


FIGURE 8 – Dans un dictionnaire chaque élément possède une étiquette.

Représentation des données types construits

Surmonter les difficultés

Représentation des données types construits

Les programme

En première

. . . .

Ambiguït

Lableau

Delinition

Complexité temporell

Définition

Complexité tem

ist Python

arcours de listes dictionnaires

Difficultés rencontrées

Surmonter les difficultés

Fonction et

onction et Iramètre mutable

les, files

Définition

- 4. Fonction et paramètre mutable

Représentation des données types construits

Fonction et paramètre mutable

Fonction et paramètre mutable

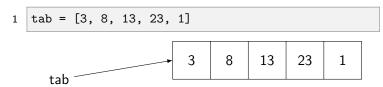


FIGURE 9 – La variable tab contient une référence au tableau.

Représentation des données types construits

Les programmes

En première

Lietoe

Ambiguïté

Fableau

Complexité temporelle

Liste chaînée

Définition

Complexité tempore

Parcours de listes

et dictionnaires

Difficultés rencontrées Surmonter les difficultés

Surmonter les difficultés Fonction et

paramètre mutable

Piles, files

Définition

Interface

tab = [3, 8, 13, 23, 1] tab1 = tab

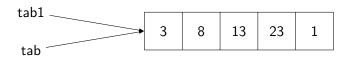


FIGURE 10 – La ligne 2 ne copie pas le contenu du tableau mais sa référence.

Représentation des données types construits

Les programmes

En première

ictec

Ambiguïté

bleau

E-tat--

nplexité temporelle

Définition

- .

list Python

Parcours de listes et dictionnaires

Difficultés rencontrées Surmonter les difficulté

Fonction et

paramètre mutable

Piles, files

Définition

Interface

```
def ma_fonction(t: list) -> None:
    t[0] = 99

tab = [3, 8, 13, 23, 1]
ma_fonction(tab)
```

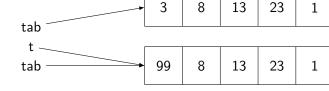


FIGURE 11 – Le paramètre t contient une référence au tableau.

Visualiser sur Pythontutor

Représentation des données types construits

Les programmes

in première

istes

Ambiguïté

ableau

Affinition

Complexité temporelle

te chaînée

Définition

Complexite te

Parcoure de lie

et dictionnaires

Surmonter les difficulte

Fonction et paramètre mutable

Piles, files

Définition

Interrace

```
def prefixe(arbre: list, i: int = 0, p: list = []):
    if i < len(arbre):</pre>
        p.append(arbre[i])
        prefixe(arbre, 2*i+1, p)
        prefixe(arbre, 2*i+2, p)
    return p
arbre = [0, 1, 2, 3, 4, 5, 6]
parcours = prefixe(arbre)
parcours = prefixe(arbre)
```

4

5 6

8

9

10 11

Code 2 – valeur par défaut dans paramètre

es programmes

En première

....

mbiguïté

bleau

r ...

mplexité temporelle

iste chaînée

Définition

Complexité te

ist Python

Parcours de listes et dictionnaires

Difficultés rencontrées Surmonter les difficultés

Fonction et paramètre mutable

iles files

Piles, files

Interface

Représentation des données types construits

1. Les programmes

2. Liste:

3. Parcours de listes et dictionnaires

4. Fonction et paramètre mutable

5. Piles, files

5.1 Définition

5.2 Interface

5.3 Implémentations

Les programmes

En première

. . .

Listes

Ambiguïté

Tableau

Definition

Complexité temporelle

Liste Chainee

Définition

Complexite t

ist Python

Parcours de listes

t dictionnaires

Difficultés rencontrées

Surmonter les difficultés

onction et aramètre m

Piles, files

Définition

Interface

Représentation des données types construits

5. Piles, files

5.2 Interface

Interface

Représentation des données types construits

1. Les programmes

2. Listes

3. Parcours de listes et dictionnaires

4. Fonction et paramètre mutable

5. Piles, files

5.1 Définition

5.2 Interface

5.3 Implémentations

Les programmes

En première

. . .

istes

Ambiguïté

Tableau

Definition

Complexité temporelle

Liste chainee

Définition

Comployitá tom

list Python

Parcours de listes

t dictionnaires

Difficultés rencontrées

Surmonter les difficultés

oaramètre muta

Piles, files

Définition

Interface