

Types abstraits et structures de données

François Schwarzentruher
ENS Cachan – Antenne de Bretagne

Type abstrait = Besoin

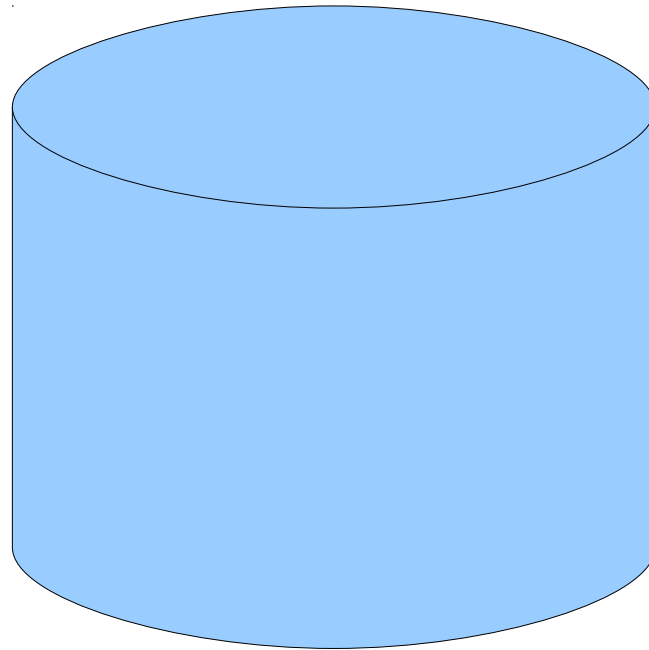
Données : Ensemble

Opérations :

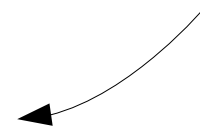
- Ajouter un élément
- Supprimer
- Tester l'appartenance

Type abstrait `Ensemble`

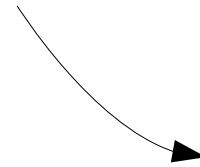
Tester
l'appartenance



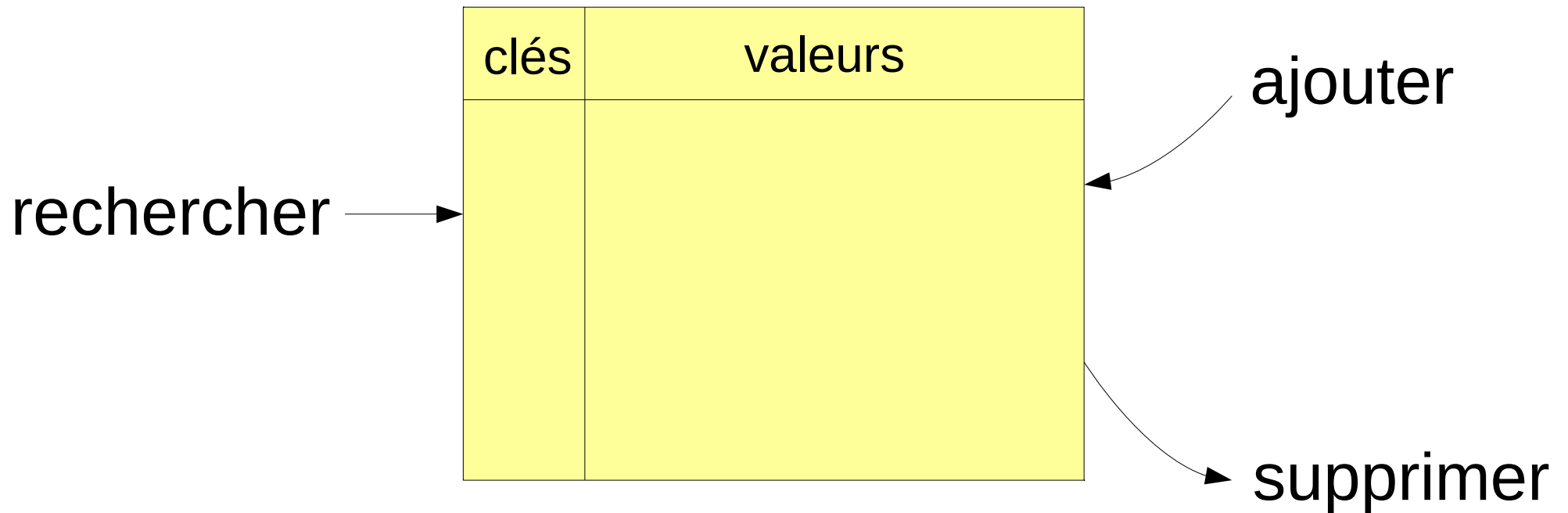
ajouter



supprimer



Type abstrait : Tableau associatif



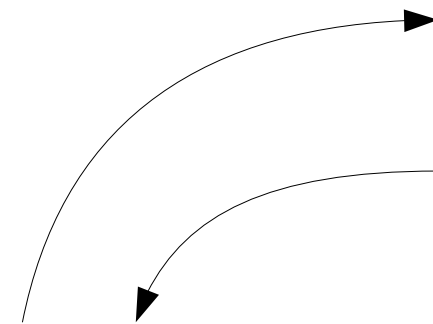
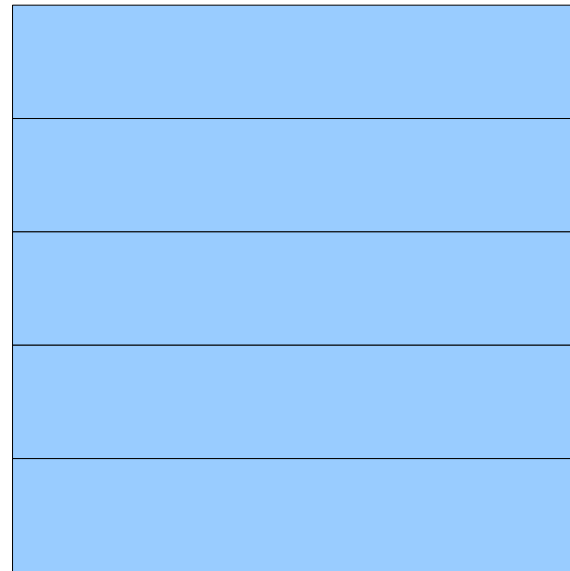
Types abstraits...

- **Relation d'équivalence**
fusionner des classes d'équivalences, tester si $x \sim y...$
- **N-uplets**
insérer à la i^e coordonnée

Pile (LIFO : last in, first out)

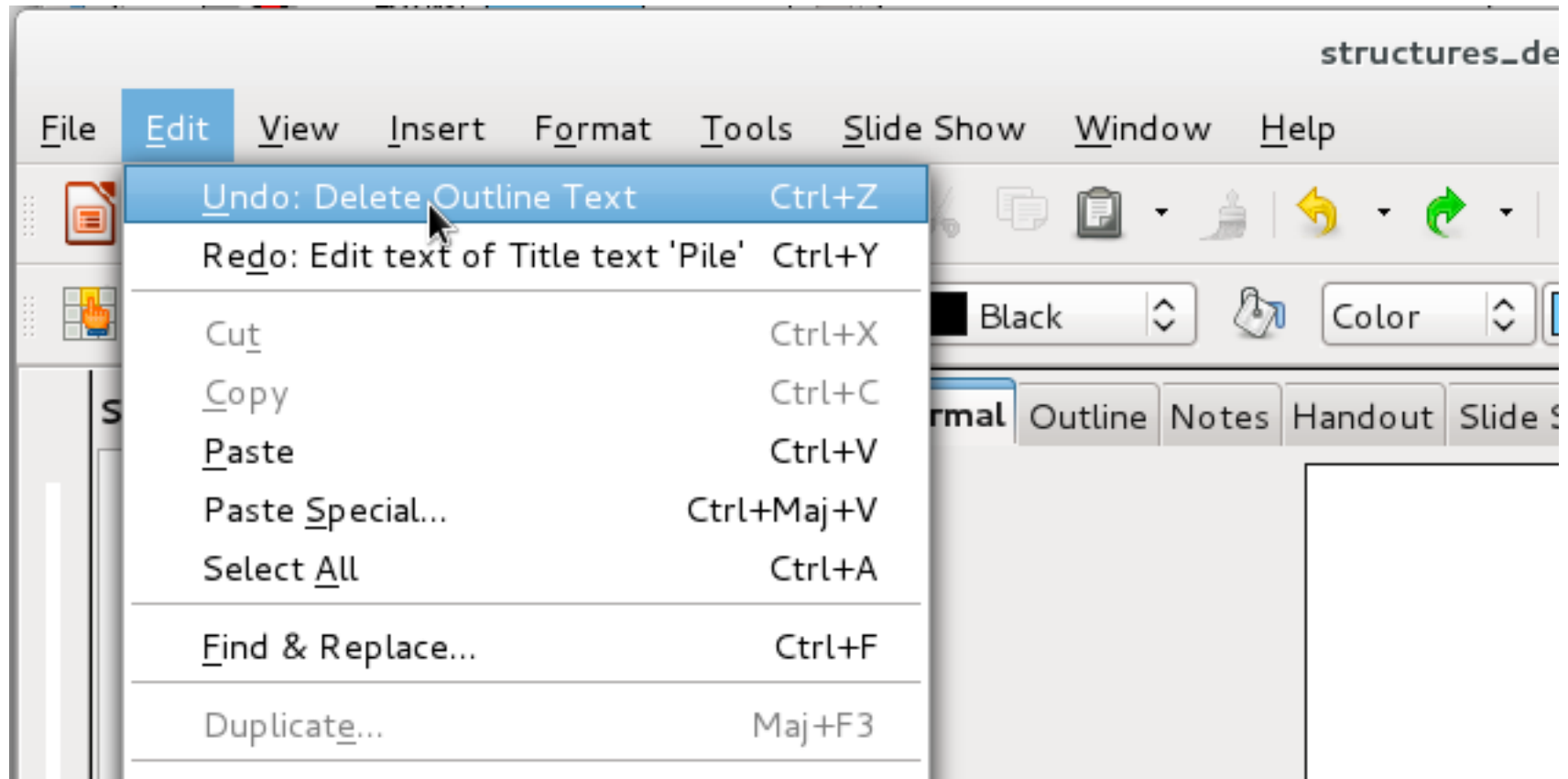


EstVide?



dépiler
empiler

Fonctions annuler/refaire



Pile d'appel

procedure fractale(*n*)

if *n* = 0

| *dessiner un trait*

else

| fractale(*n* - 1)

● *tourner de 30 degrés*

| fractale(*n* - 1)

● *tourner de -60 degrés*

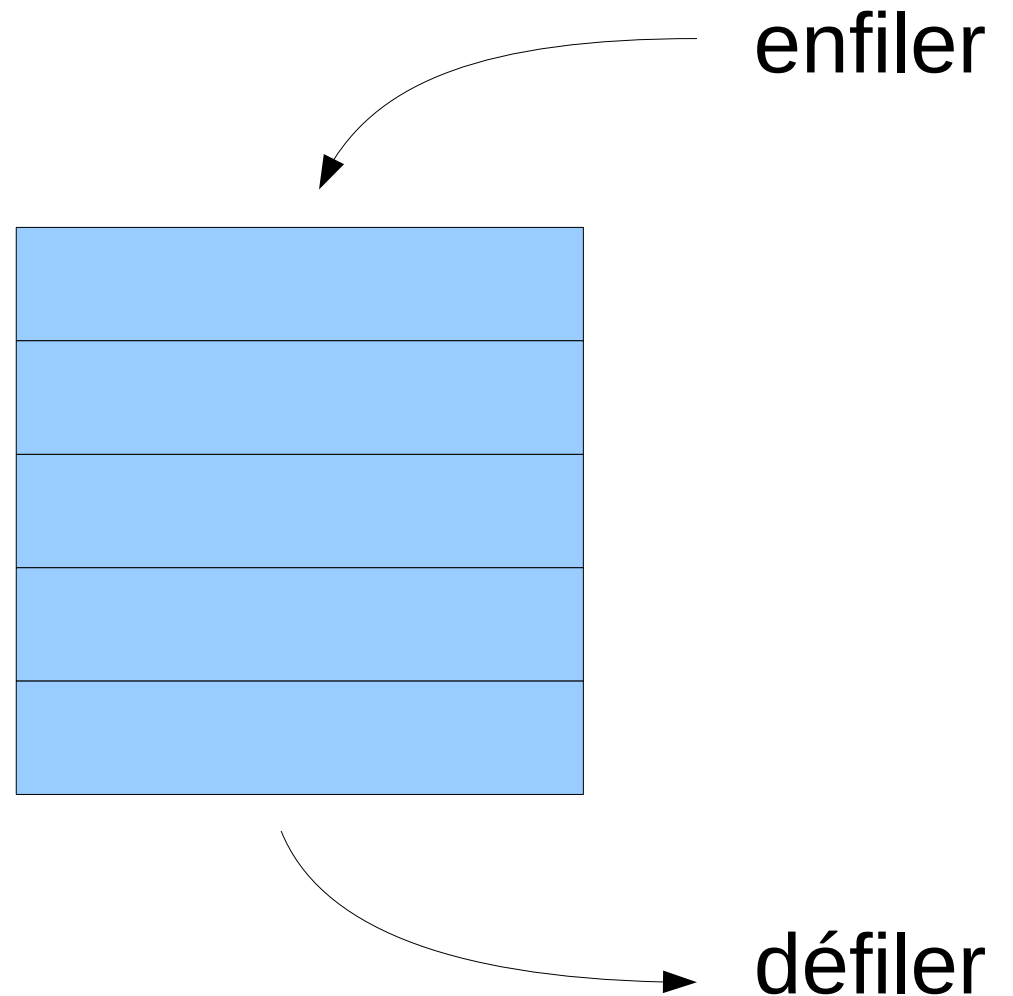
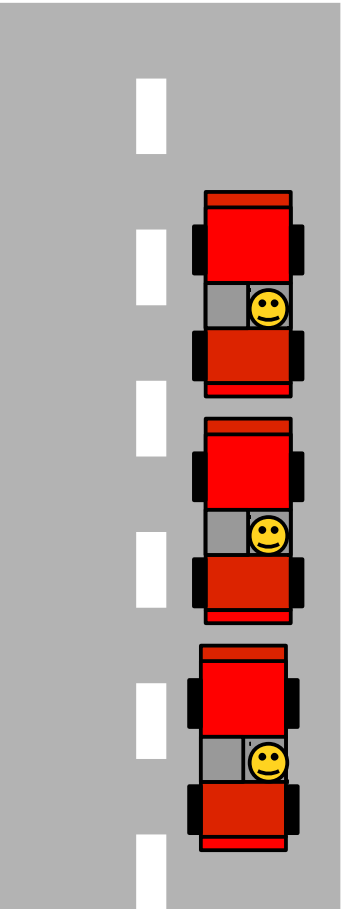
| fractale(*n* - 1)

● *tourner de 30 degrés*

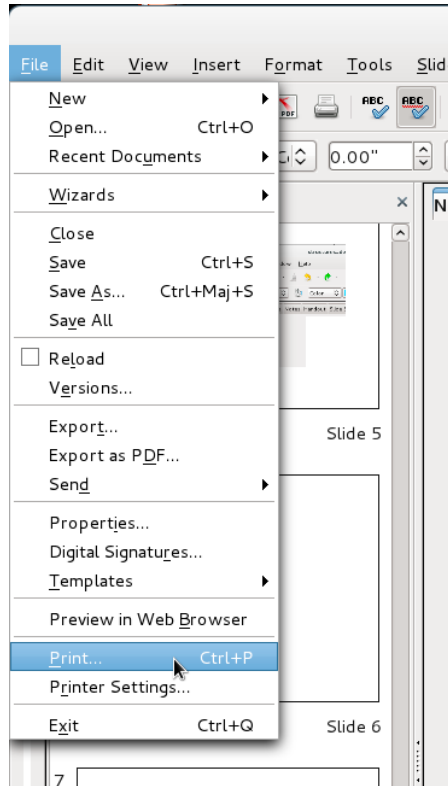
| fractale(*n* - 1)

● *endFunction*

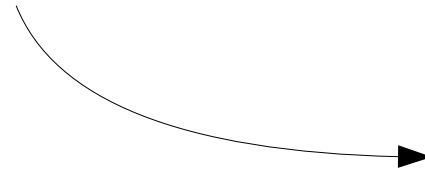
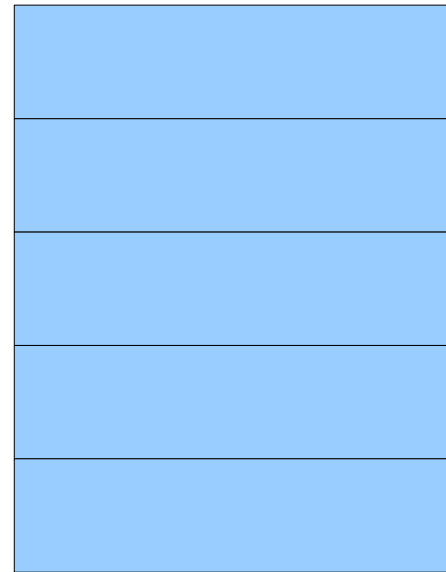
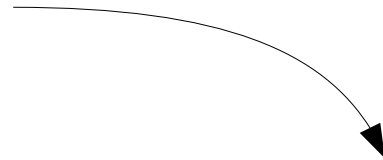
File (FIFO : first in, first out)



Gestionnaire d'une imprimante



enfiler



défiler

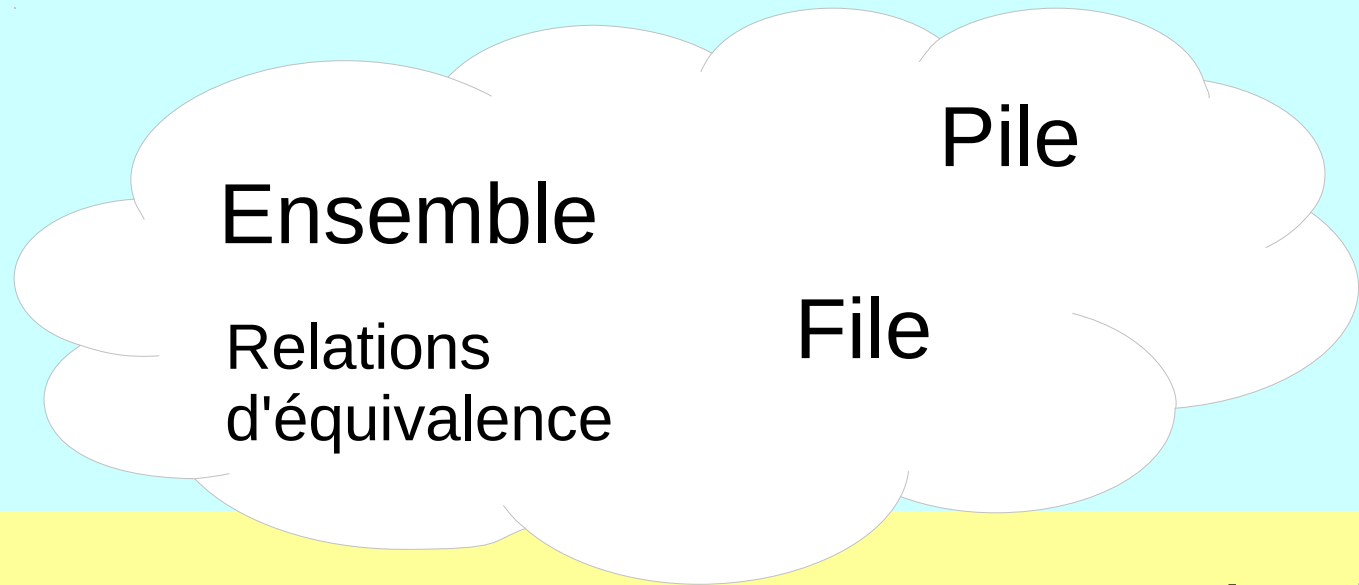


Paysage général

Problème et un algorithme

Besoins

= type abstrait
= interface



Moyens

= structure de données
= implémentation

Tableau

Arbres

Liste

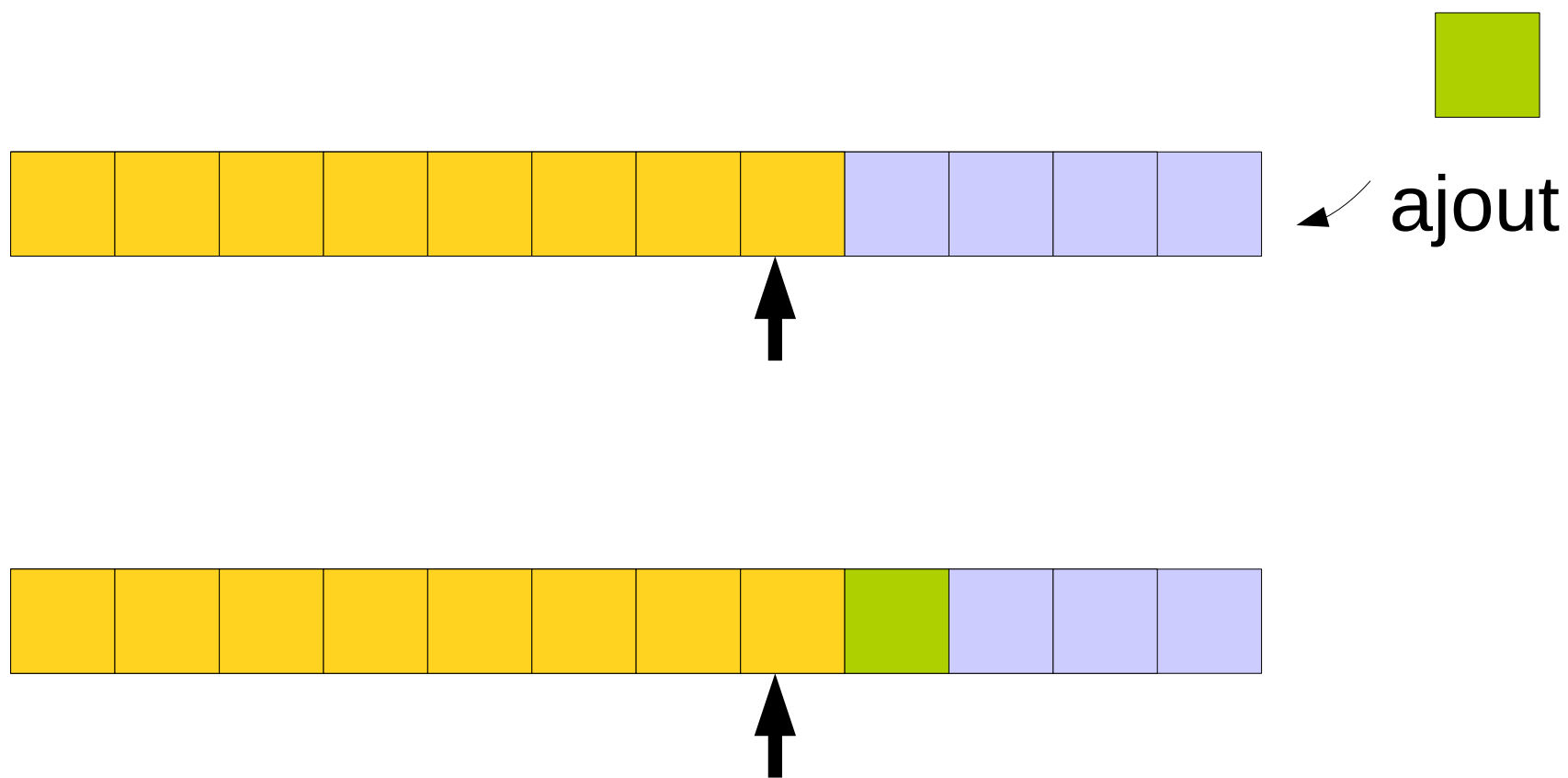
Table de hachage



Implémentations



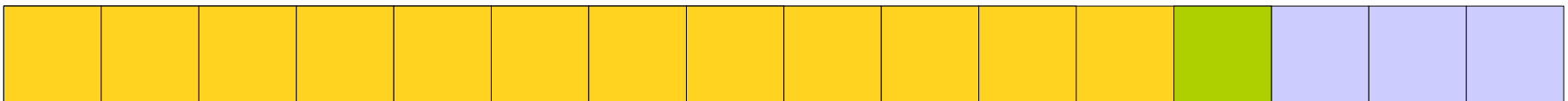
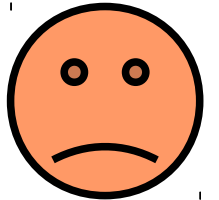
Ajout dans un tableau



Ajout dans un tableau

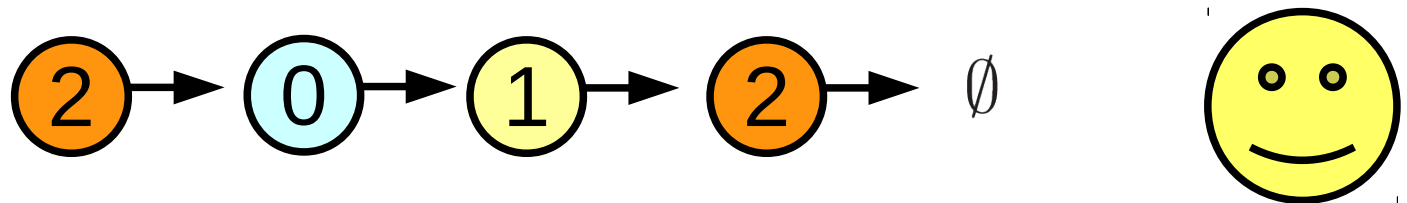
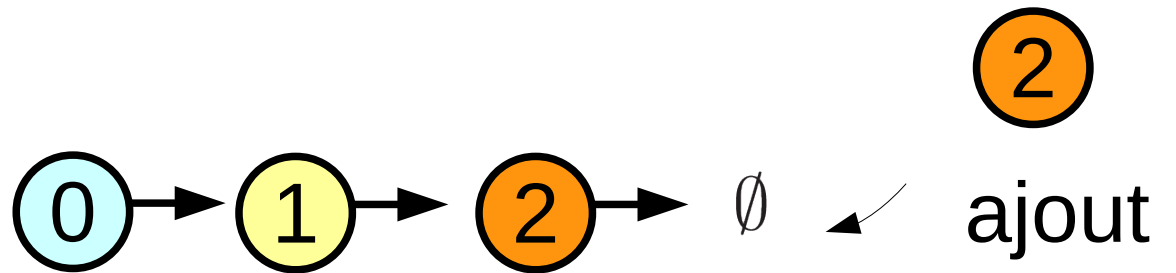


↙ ajout



allocation mémoire

Ajout dans une liste



Etude de cas(L)

\emptyset

...

$e \rightarrow L$

...

match(L)

\emptyset

...

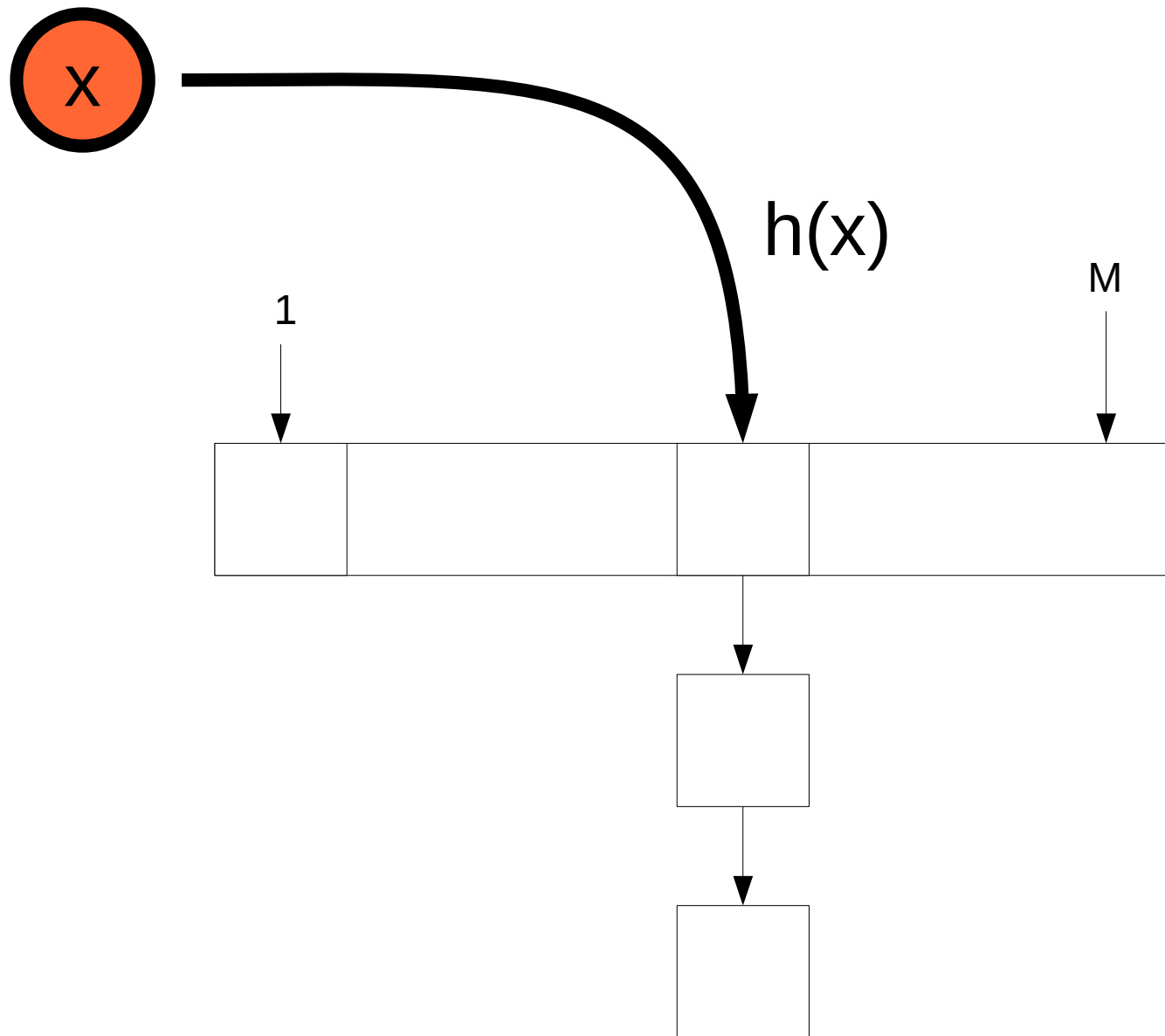
$e \rightarrow L$

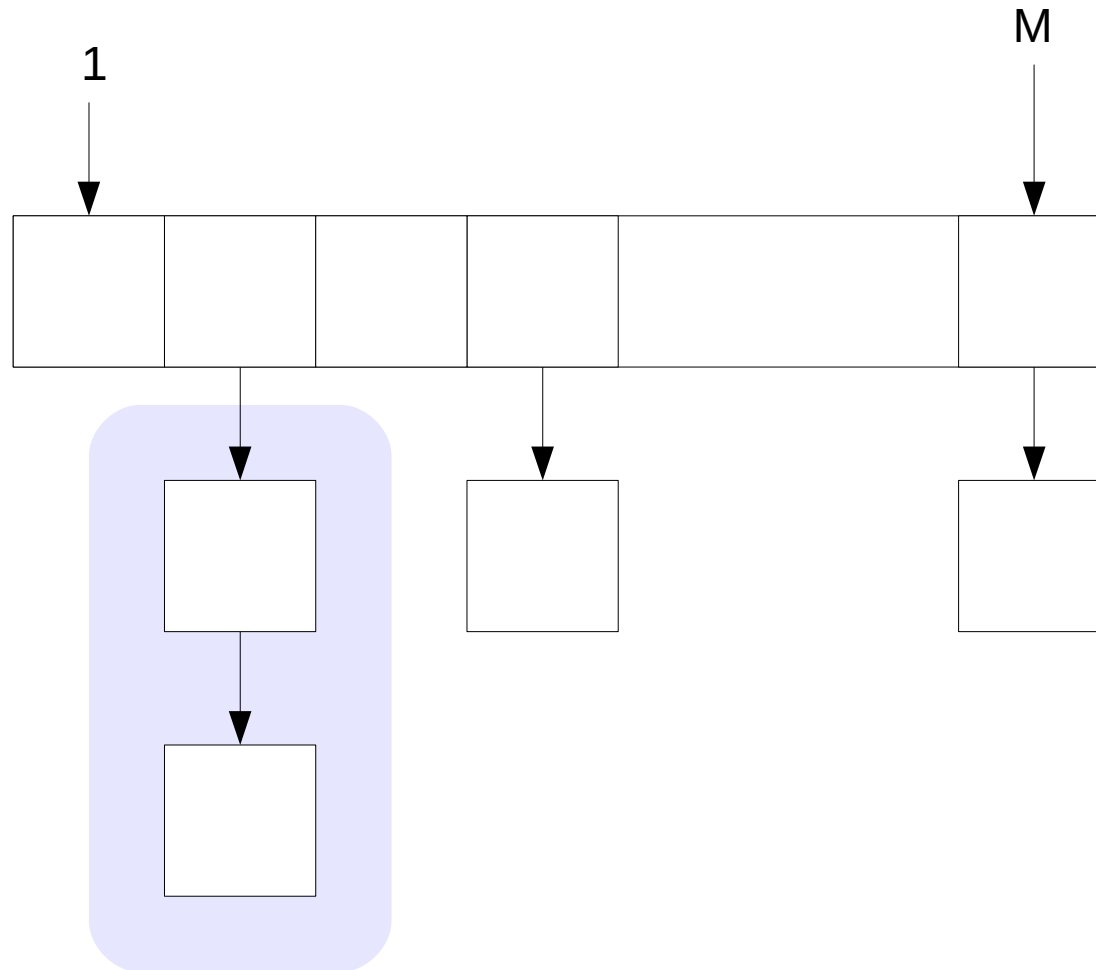
...



Table de hachage

Table de hachage

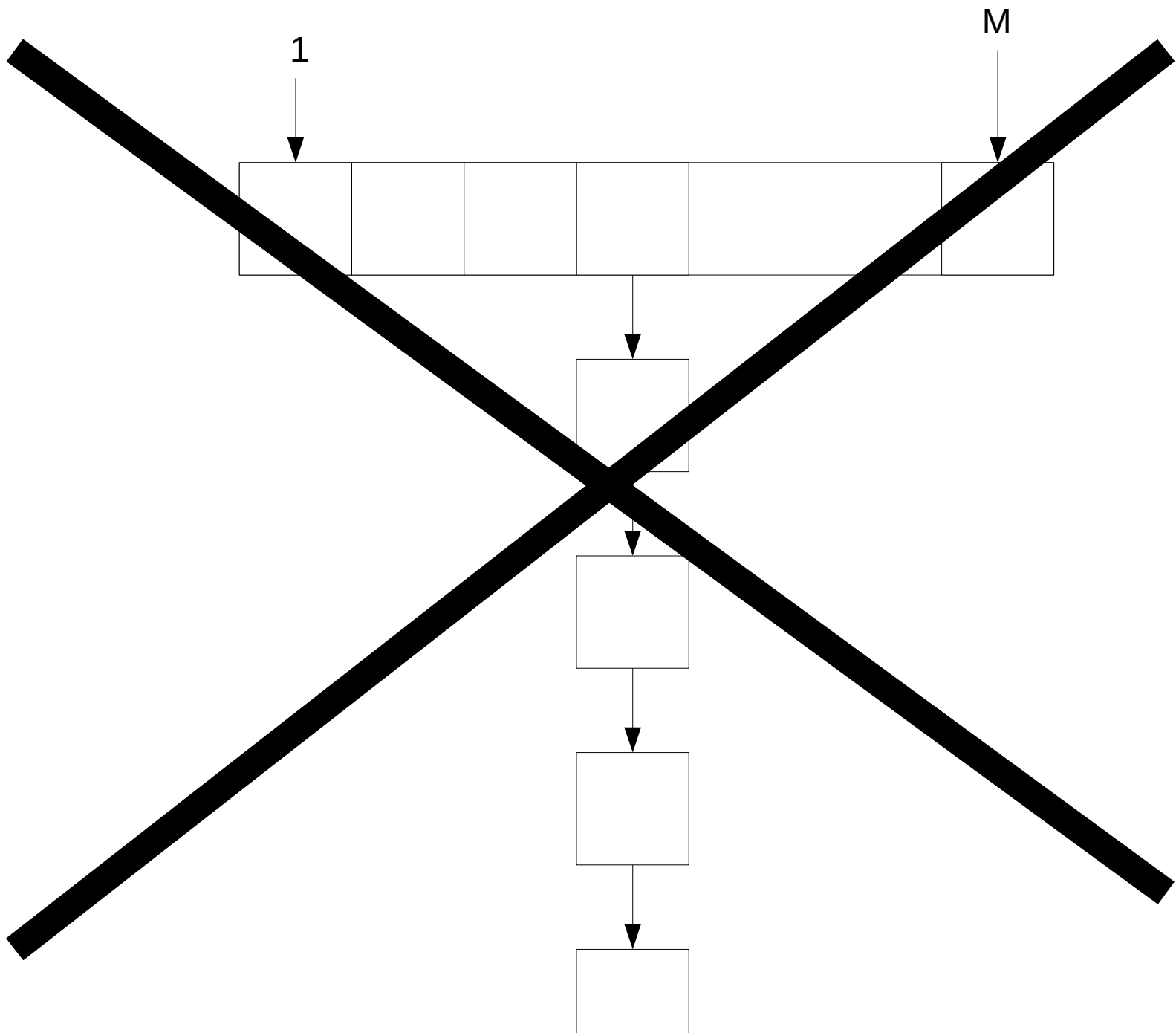




colision !

Nos désirs

- Peu de collisions
- M petit
- h rapide à calculer



Hachage uniforme

