

TD2: ASD

Exercice 3 : Implantation de files

On souhaite implanter une file avec une liste simplement chaînée. On suppose avoir accès aux primitives et aussi à la structure de donnée elle-même, ce qui signifie qu'on peut modifier la liste sans passer par les primitives.

Q 3.1 Comment réaliser l'opération **enfiler** sans utiliser de boucle implicitement ou explicitement ?

fonc **enfiler** : lst el :
el.next ← lst
ret el # el devient la tête de la liste (Ø wrapper)

On change maintenant notre représentation de file et on s'autorise à encapsuler la liste dans une autre structure de données afin de pouvoir réaliser l'opération **défiler** sans utiliser de boucle implicitement ou explicitement.

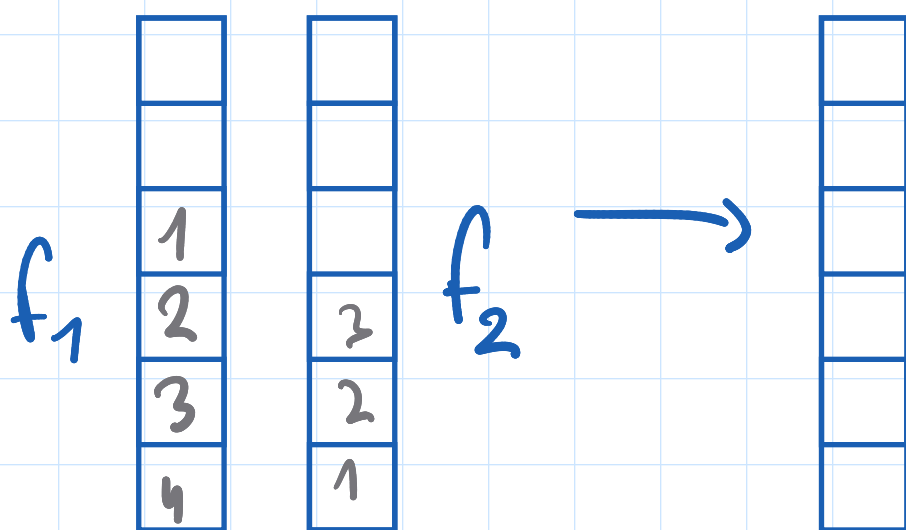
Q 3.2 Quelle information faut-il conserver pour garantir cela ?

On stocke la liste à l'envers, et on garde head et tail
→ 4 3 2 1 →

Exercice 4 : Files et piles

En utilisant uniquement les opérations primitives sur chacune des deux structures de données.

Q 4.1 Montrer que l'on peut implanter une pile avec deux files.



empiler → enfiler dans f_1
dépiler → défiler f_1 dans f_2 jusqu'à ce que la taille de la file soit 1
défile f_1
défile f_2 dans f_1

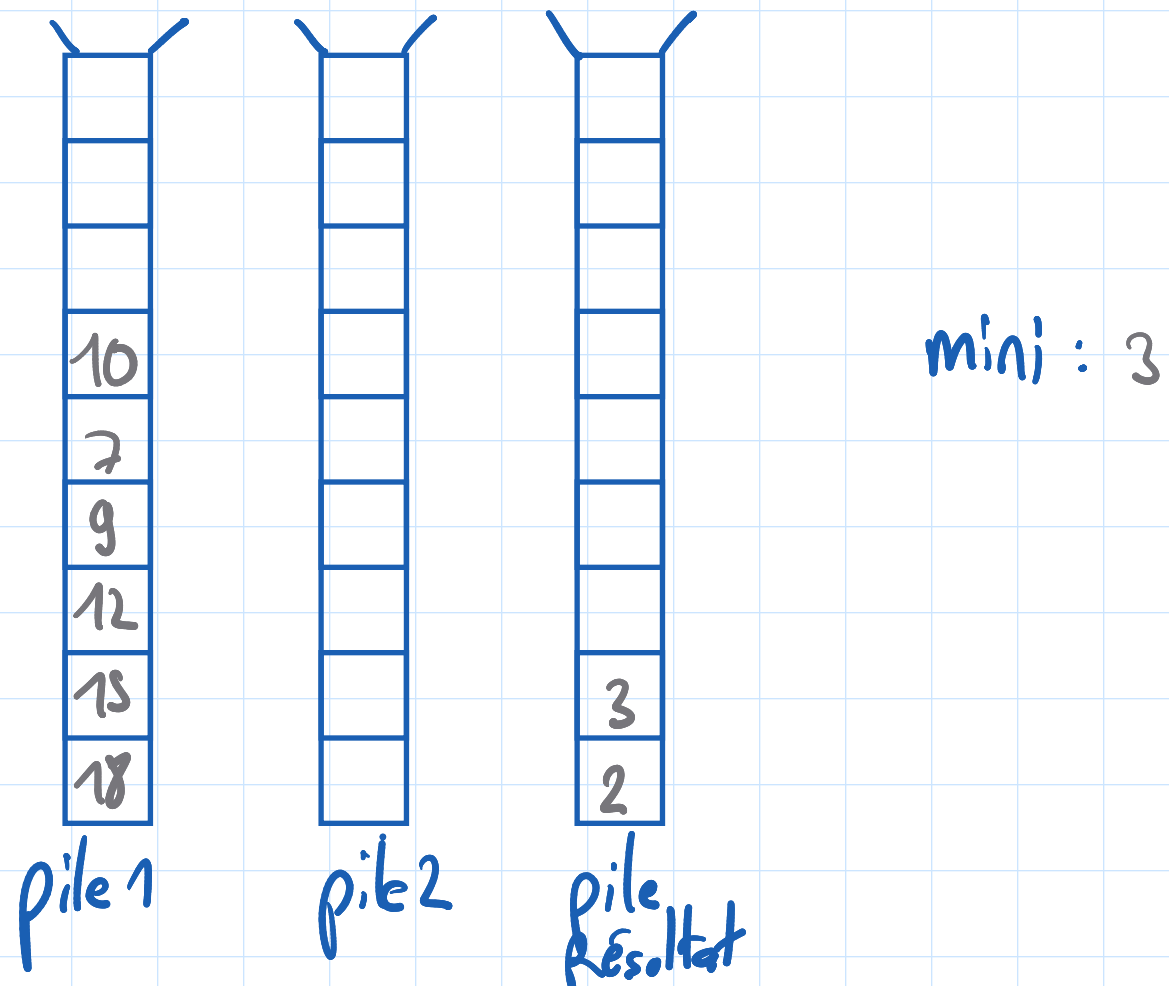
Exercice 5 : Tri par sélection avec des piles

On rappelle le principe du tri par sélection du minimum :

```
soit E une suite d'elements a trier
soit F une suite vide
tant qu'il reste des elements dans E
  soit x l'element minimum de E
  ranger x dans F (de maniere ordonnee)
fin tant que
retourner F
```

On souhaite utiliser cet algorithme pour trier les éléments d'une pile (le plus petit élément sera au fond de la pile et le plus grand au sommet). La pile est la seule structure de données autorisée dans cet exercice (on n'aura droit ni aux tableaux, ni aux files, ni aux listes). Ainsi votre algorithme pourra utiliser d'autres piles et bien sûr des variables.

Q 5.1 Décrire un algorithme en français réalisant le tri des éléments d'une pile utilisant le principe du tri par sélection du minimum.



tant que pile 1 pas vide

Dépiler pile 1 dans pile 2 en conservant le minimum

Dépiler pile 2 dans pile 1 jusqu'au minimum

Dépiler le minimum dans pile Résultat

Dépiler le reste de pile 2 dans pile 1