

6.5.2 Les files

Les opérations sur les files sont

1. fabriquer la file vide,
2. tester si une file est vide,
3. ajouter un dernier élément à une file,
4. accéder au premier élément de la file,
5. retirer le premier élément de la file.

La seule différence avec une pile est donc que l'on ajoute un élément à la fin d'une file alors que l'on ajoute un élément au début d'une pile : premier arrivé, premier servi. Les files sont aussi parfois appelées *fifo* : *first in, first out*.

Exercice 6.15

Programmer les opérations ci-dessus en représentant les files comme des listes enveloppées. Combien de temps demande l'opération d'ajout d'un élément ?

Exercice 6.16

Pour rendre l'opération d'ajout d'un élément plus rapide, on peut représenter une file dont on connaît *a priori* la taille maximale \max par un tableau de taille \max . Les éléments d'une file de taille n sont mis dans les cases $i \% \max$, $(i + 1) \% \max$, ... $(i + n - 1) \% \max$ pour un entier i quelconque.

Programmer les opérations ci-dessus sur cette structure de données.

Exercice 6.17

Pour rendre l'opération d'ajout d'un élément plus rapide, on peut également représenter une file par deux listes. Le couple formé des listes l_1, \dots, l_p et m_1, \dots, m_q représentant la file $l_1, \dots, l_p, m_q, \dots, m_1$.

Programmer les opérations ci-dessus sur cette structure de données.

Combien de temps demande l'opération d'accès au premier élément les jours de chance ? Et dans le pire des cas ? Et en moyenne ?

6.5.3 Les files de priorité

Au service des urgences d'un hôpital, la règle du premier arrivé, premier servi n'est pas toujours optimale. Une meilleure solution est d'attribuer à