

---

**Algorithme 1** Fonction Insérer( $x, Q$ )

---

**Entrée :**  $Q$  queue avec priorité représentée par un tas.

```
 $Q.dernier := Q.dernier + 1;$   
 $Q.tas[Q.dernier] := x;$   
Monter( $Q.dernier, Q$ )
```

---

---

**Algorithme 2** Fonction Monter( $p, Q$ )

---

```
si  $p > 1$  alors  
  si  $Q.tas[p] < Q.tas[p \text{ div } 2]$  alors  
    Echanger( $Q.tas[p], Q.tas[p \text{ div } 2]$ );  
    Monter( $p \text{ div } 2, Q$ )  
  fin si  
fin si
```

---

---

**Algorithme 3** Fonction EnleverMin( $Q$ )

---

**Entrée :**  $Q$  queue avec priorité représentée par un tas.

```
 $Q.tas[1] := Q.tas[Q.dernier];$   
 $Q.dernier := Q.dernier - 1;$   
Descendre(1,  $Q$ )
```

---

---

**Algorithme 4** Fonction Descendre( $p, Q$ )

---

```
si  $p \leq Q.dernier \text{ div } 2$  alors  
   $r := 2 * p;$   
  si  $r < Q.dernier$  alors  
    si  $Q.tas[r] > Q.tas[r + 1]$  alors  
       $r := r + 1$   
    fin si  
  si  $Q.tas[p] > Q.tas[r]$  alors  
    Echanger( $Q.tas[p], Q.tas[r]$ );  
    Descendre( $r, Q$ )  
  fin si  
fin si  
fin si
```

---

Complexité d'Insérer et EnleverMin en  $\mathcal{O}(\log n)$  où  $n$  est le nombre d'éléments dans  $Q$ .

---

**Algorithme 5** Fonction Entasser( $Q$ )

---

**Entrée :**  $Q$  tableau quelconque.

**Sortie :**  $Q$  tas.

```
pour  $i$  décroissant de  $Q.dernier \text{ div } 2$  à 1 faire  
  Descendre( $i, Q$ )  
fin pour
```

---

Complexité d'Entasser en  $\mathcal{O}(n)$ .