

Heimadæmi – heimadæmi 9 hópverkefni

Arnar Sigurðsson

```
{;;;
```

Hönnunarskjal fyrir Pqueue einingu sem gefur
okkur kost á að vinna með fjölnota forgangsbiðröð.

Útflutt úr einingunni

=====

Notkun: `b = makePqueue()` ;

Fyrir: Ekkert.

Eftir: `b` inniheldur nýja tóma forgangsbiðröð af ótakmarkaðri stærð.

Notkun: `addToPqueue(b,x,l)` ;

Fyrir: `b` er forgangsbiðröð (fengin úr `makePqueue`).

`x` er gildið sem á að bæta við forgangsbiðröð.

`l` er lykill

Eftir: heiltölunni `x` hefur verið bætt í forgangsbiðröðina `b`.

Notkun: `e = isEmpty(b)` ;

Fyrir: `b` er forgangsbiðröð.

Eftir: `e` er satt þpaa `b` sé tóm.

Notkun: `x = removeFromPqueue(b)` ;

Fyrir: `b` er forgangsbiðröð, ekki tóm.

Eftir: Búið er að fjarlægja stak úr forgangsbiðröð.

Þetta stak er gildi úr forgangsbiðröðinni

sem hefur lykil með mestan forgang.

`x` er stakið sem var fjarlægt.

```
;;;}
```

```

"Pqueue.mmod" =
{{

;;; Fastayrðing gagna:
;;;   Poki para af gildum þar sem annað gildið er
;;;   tákn (t.d. heiltala, strengur) og hitt er lykillinn af því
;;;   tákni sem leyfir samanburð á milli þessa
;;;   tákns og annars í pokaum. Í pokaum eru
;;;   semsagt pör [x1, l1],[x2, l2],...,[xN, lN] og
;;;   pokinn er táknaður sem bendir á
;;;   falda breytu x þar sem x er
;;;   listi af listum [[x1, l1],[x2, l2],...,[xN, lN]],
;;;   eða listi af listum sömu para í annarri röð þar sem
;;;   hver listi í listanum inniheldur tákn og lykil þess tákns.

makePqueue =
    fun()
    {
        var x = [];
        &x;
    };

addToPqueue =
    fun(q,x,l)
    {
        var temp = x:l;
        *q = temp : *q;
    };

isEmpty =
    fun(q)
    {
        ! *q
    };

```

```

removeFromPqueue =
  fun(q)
  {
    var temp = *q;
    var min = tail(head(temp));
    var skilagildi = head(temp);
    while(temp!=null){
      ;; temp er hlutmengi í q þar sem
      ;; lengd á lista:  $0 \leq \text{temp} \leq *q$ 
      if(tail(head(temp)) <<>> min){
        min = tail(head(temp));
        skilagildi = head(temp);
      };
      temp = tail(temp);
    };
    temp = *q;
    var nyrListi = [];
    while(skilagildi != head(temp)){
      ;; heildarlengd nyrListi + temp = lengd *q
      ;; eða upphaflegi listinn.
      var stak = head(temp);
      nyrListi = stak : nyrListi;
      temp = tail(temp);
    };
    if(tail(temp) == null){
      *q = nyrListi;
      skilagildi;
    };
    temp = tail(temp);
    while(temp != null){
      ;; heildarlengd nyrListi + temp = lengd *q - 1
      ;; eða upphaflegi listinn án staks sem var
      ;; hoppað yfir.
      var stak = head(head(temp)) : tail(head(temp));
      nyrListi = stak : nyrListi;
      temp = tail(temp);
    };
    *q = nyrListi;
    skilagildi;
  };
}}
*
{{
<<>> =
  fun(x,y)
  {
    var out = x<<<y;
  };
}}
;

```

```

"testPqueue1.mex" = main in
{{
  ;;; Notkun: main();
  ;;; Fyrir: Ekkert.
  ;;; Eftir: Búið er að skrifa tölurnar 1-5 í vaxandi röð,
  ;;;        þá 1-2-3-4-5.
  ;;;
  main =
    fun()
    {
      ;;; Prófun 1
      writeln();
      writeln("PROFUN 1");
      writeln();
      writeln("creating empty priorityQueue");
      var x = makePqueue();
      write("isEmpty(): "); writeln(isEmpty(x));
      writeln("adding 5 elements to the priorityQueue");
      addToPqueue(x, 1, 1);
      addToPqueue(x, 5, 5);
      addToPqueue(x, 4, 4);
      addToPqueue(x, 3, 3);
      addToPqueue(x, 2, 2);
      write("done adding elements, checking if isEmpty(): "); writeln(isEmpty(x));
      writeln("removing all 5 elements from priorityQueue and printing results:");
      for(var i = 1; i<6; i = i + 1){
        var utkoma = removeFromPqueue(x);
        writeln(tail(utkoma));
      };
      writeln("done removing elements");
      write("priorityQueue isEmpty(): "); writeln(isEmpty(x));
    };
  }}
*
"Pqueue.mmod"
*
{{
  <<< = fun <(x,y);
  }}
*
BASIS
;

```

```

"testPqueue2.mexe" = main in
{{
  ;;; Notkun: main();
  ;;; Fyrir: Ekkert.
  ;;; Eftir: Búið er að skrifa tölurnar 1-5 í minnkandi röð,
  ;;;         þá 5-4-3-2-1.
  ;;;
  main =
    fun()
    {
      ;;; Prófun 2
      writeln();
      writeln("PROFUN 2");
      writeln();
      writeln("creating empty priorityQueue");
      var x = makePqueue();
      write("isEmpty(): "); writeln(isEmpty(x));
      writeln("adding 5 elements to the priorityQueue");
      addToPqueue(x, 1, 1);
      addToPqueue(x, 5, 5);
      addToPqueue(x, 4, 4);
      addToPqueue(x, 3, 3);
      addToPqueue(x, 2, 2);
      write("done adding elements, checking if isEmpty(): "); writeln(isEmpty(x));
      writeln("removing all 5 elements from priorityQueue and printing results:");
      for (var i = 1; i < 6; i = i + 1) {
        var utkoma = removeFromPqueue(x);
        writeln(tail(utkoma));
      };
      writeln("done removing elements");
      write("priorityQueue isEmpty(): "); writeln(isEmpty(x));
    };
  }}
*
"Pqueue.mmod"
*
{{
  <<< = fun >(x,y);
  }}
*
BASIS
;

```

```
C:\Users\addi\Desktop\Háskóli\onn 3\Forritunarmal\Verkefni 8>java -jar morpho.jar testPqueue1
```

```
PROFUN 1
```

```
creating empty priorityQueue  
isEmpty(): true  
adding 5 elements to the priorityQueue  
done adding elements, checking if isEmpty(): false  
removing all 5 elements from priorityQueue and printing results:  
1  
2  
3  
4  
5  
done removing elements  
priorityQueue isEmpty(): true
```

```
C:\Users\addi\Desktop\Háskóli\onn 3\Forritunarmal\Verkefni 8>java -jar morpho.jar testPqueue2
```

```
PROFUN 2
```

```
creating empty priorityQueue  
isEmpty(): true  
adding 5 elements to the priorityQueue  
done adding elements, checking if isEmpty(): false  
removing all 5 elements from priorityQueue and printing results:  
5  
4  
3  
2  
1  
done removing elements  
priorityQueue isEmpty(): true
```