

Volem representar cues de forma que millorem l'eficiència conjunta de les operacions d'afegir i d'avançar. Per això, implementem la cua amb dues llistes tals que si concatenem la primera amb la revessada de la segona tenim els elements de la cua en l'ordre de sortida. Usant com a constructor *Queue*, un exemple de cua seria

```
cua = Queue [2,8,5] [4,7]
```

que representa la cua on el primer és el 2 i segueix amb 8, 5, 7 i 4.

D'aquesta manera, l'operació d'afegir en una cua es fa posant el nou element per davant de la segona llista (que és menys costós que afegir-lo al final d'una llista).

D'altra banda, l'operació d'avançar es fa treient el primer de la primera llista, si en té, i sinó, passant els de la segona llista cap a la primera (en l'ordre correcte) i agafant el primer tot deixant la resta.

1. Implementeu cues genèriques utilitzant la definició de dades i les operacions següents:

```
data Queue a = Queue [a] [a]
deriving (Show)
```

```
create :: Queue a
push :: a → Queue a → Queue a
pop :: Queue a → Queue a
top :: Queue a → a
empty :: Queue a → Bool
```

2. Definiu la igualtat de cues de manera que dues cues siguin iguals si i només si tenen els mateixos elements en el mateix ordre de sortida. Per a fer-ho, indiqueu que les cues són una instància de la classe **Eq** on (**==**) és l'operació d'igualtat que heu de definir:

```
instance Eq a ⇒ Eq (Queue a)
where
  ...
```

Fixeu-vos que cal que el tipus dels elements de la cua també siguin de la classe **Eq**.

Puntuació

Cada apartat puntua 50 punts.

Exemple d'entrada 1

```
let c = push 3 (push 2 (push 1 create))
c
top c
pop c
empty $ pop c
empty $ pop $ pop $ c
empty $ pop $ pop $ pop c
```

Exemple de sortida 1

```
Queue [] [3,2,1]
1
Queue [2,3] []
False
False
True
```

Exemple d'entrada 2

```
let c1 = push 4 (pop (push 3 (push 2 (push 1 create))))
let c2 = push 4 (push 3 (push 2 create))
c1
c2
c1 == c2
```

Exemple de sortida 2

```
Queue [2,3] [4]
Queue [] [4,3,2]
True
```

Informació del problema

Autor : Jordi Petit
Generació : 2024-05-03 00:00:17

© *Jutge.org*, 2006–2024.
<https://jutge.org>