

演習問題 8.2

回転操作をはじめる時に 2 回、他の挿入や削除では 1 回ずつ exec を呼び出せば回転操作完了を間に合わせるには十分であることを示せ

注意) 厳密な証明ではなく、連続して snoc, tail を呼ぶ場合のみを考えている。

回転開始時点での、f, r の長さをそれぞれ m, m+1 とすると、回転完了には exec が $2m+2$ 回必要である。

snoc

連続して snoc するとき、キューの長さは $2m+1$ と表すことができる。このとき、次の回転までに $2m+2$ ステップ猶予がある。

- 例えば、キューの長さが 7 ($|f|=3$) のとき、snoc を繰り返して次の回転が起きるのは長さ 15 のときである。つまり 8 ステップ猶予があり、 $8 = (2|f|+2)$ と表せる。(雑)

この場合、各 snoc で 1 回ずつ exec を呼べば間にあう

tail

逆に、連続して tail するとき、次の回転までに m ステップ猶予がある。(fの長さの分)

この間に、長さ m+1 である r の回転を完了させなければならないので、m ステップで exec を m+1 回呼ぶことになる。

- この場合、回転開始に 2 回、あとは 1 回ずつ exec を呼べば間にあう

観察

特に意味は無いです

```
-- mapM_ (print . foldl snoc empty) [[1..x] | x <- [1..13]]
mapM_ print $ scanl snoc empty [1..13] -- こっちのほうが簡単

Q 1 [1] Idle 0 []
Q 1 [1] Idle 1 [2]
-- check (Q 1 [1] Idle 2 [3:2])
-- Q 3 [1] (exec (exec (Reversing 0 [1] [] [3,2] []))) 0 []
-- Q 3 [1] (exec (Reversing 1 [] [1] [2] [3])) 0 []
Q 3 [1] (Appending 1 [1] [2,3]) 0 []
Q 3 [1] (Appending 0 [] [1,2,3]) 1 [4]
Q 3 [1,2,3] Idle 2 [5,4]
Q 3 [1,2,3] Idle 3 [6,5,4]
-- check (Q 3 [1,2,3] Idle 4 [7,6,5,4])
-- Q 7 [1,2,3] (exec (exec (Reversing 0 [1,2,3] [] [7,6,5,4] []))) 0 []
-- Q 7 [1,2,3] (exec (Reversing 1 [2,3] [1] [6,5,4] [7])) 0 []
Q 7 [1,2,3] (Reversing 2 [3] [2,1] [5,4] [6,7]) 0 []
Q 7 [1,2,3] (Reversing 3 [] [3,2,1] [4] [5,6,7]) 1 [8]
Q 7 [1,2,3] (Appending 3 [3,2,1] [4,5,6,7]) 2 [9,8]
Q 7 [1,2,3] (Appending 2 [2,1] [3,4,5,6,7]) 3 [10,9,8]
```

```
Q 7 [1,2,3] (Appending 1 [1] [2,3,4,5,6,7]) 4 [11,10,9,8]
Q 7 [1,2,3] (Appending 0 [] [1,2,3,4,5,6,7]) 5 [12,11,10,9,8]
Q 7 [1,2,3,4,5,6,7] Idle 6 [13,12,11,10,9,8]
```

```
-- snoc 7 times -> tail
```

```
Q 7 [1,2,3] (Reversing 2 [3] [2,1] [5,4] [6,7]) 0 []
Q 6 [2,3] (Reversing 2 [] [3,2,1] [4] [5,6,7]) 0 []
Q 5 [3] (Appending 1 [3,2,1] [4,5,6,7]) 0 []
Q 4 [4,5,6,7] Idle 0 []
```