

Functional Programming

Exercises Week 8

(for November 28, 2014)

1. Read Chapter 7 of the lecture notes.

2. Exercise 7.1

3. Exercise 7.4

4. Exercise 7.8

5. Exercise 7.9

6. Implement a tail-recursive function that given a 2×2 matrix A , computes A^n .

```
mat_pow : ((int * int) * (int * int)) -> int -> ((int * int) * (int * int))
```

7. (★) Implement a priority queue with two priorities. In a queue with two priorities, each item e has two priorities associated with it $p_1(e)$ and $p_2(e)$. Apart from the insertion, it provides two extraction operations, one that finds and removes the element with maximal p_1 and one operation that finds and removes the element with maximal p_2 . The operations should be efficient, both in terms of complexity (at least $O(\log(n))$) and tail recursive.

```
empty : 'a queue  
insert : int -> int -> 'a -> 'a queue -> 'a queue  
pull_p1 : 'a queue -> ('a * 'a queue)  
pull_p2 : 'a queue -> ('a * 'a queue)
```