

# Homework 4

February 14, 2014

## 1 ML Functions with pattern matching

1. Implement a function `num_digits(n)` that takes an integer  $n$  and return the number of digits of  $n$

e.g.

```
num_digits 0 = 1
num_digits 1 = 1
num_digits 12 = 2
num_digits 123 = 3
```

2. Implement a function `max_digits(L)` that takes a list  $L$  of integers and return the maximum number of digits of the integers in  $L$ .

e.g.

```
max_digits nil = 0
max_digits [434] = 3
max_digits [12, 343, 2222, 423] = 4
```

3. Implement a function `nth_digit(x, n)` that returns  $n$ th digit of the integer  $x$  (with the least significant digit as the 1st digit). e.g.

```
nth_digit (1234, 1) = 4
nth_digit (1234, 2) = 3
nth_digit (1234, 3) = 2
nth_digit (1234, 4) = 1
```

You may assume that  $x \geq 0$  and  $n \geq 1$ .

4. Implement a function `bucket(d, n L)` that returns a sublist of integers of  $L$  so that each returned integer's  $n$ th digit is  $d$ .

e.g.

```
bucket(3, 1, [723, 234, 345, 235]) = [723]
bucket(3, 2, [723, 234, 345, 235]) = [234, 235]
bucket(3, 3, [723, 234, 345, 234]) = [345]
```

You may assume that  $0 \leq d \leq 9$  and  $n \geq 1$ .

5. Implement a function `sort_nth_digit(n, L)` that sorts the integer list  $L$  based on the  $n$ th digit of each integer. e.g.

```
sort_nth_digit(1, [723, 234, 345, 235]) = [723, 234, 345, 235]
sort_nth_digit(2, [723, 234, 345, 235]) = [723, 234, 235, 345]
sort_nth_digit(3, [723, 234, 345, 235]) = [234, 235, 345, 723]
```

You may assume that  $n \geq 1$ .

6. Implement a function `radix_sort_max(m, L)` that sorts the integer list  $L$  based on radix sort algorithm where  $m$  is the maximum number of the digits of the integers in  $L$ .

e.g.

```
radix_sort_max(3, [170, 45, 75, 90, 802, 2, 24, 66]) =
[2, 24, 45, 66, 75, 90, 170, 802]
```

You may assume that  $m \geq 1$ .

7. Implement a function `radix_sort (L)` that sorts the integer list  $L$  based on radix sort algorithm.

## 2 About radix sort

Radix sort algorithm is  $O(n)$  time sorting algorithm if we know the maximum number of digits  $m$  of the list of integers. It works by sorting the integers by their digits starting from the least significant digit. The result of the previous sort using digit  $i$  is sorted again using the digit  $i + 1$  until the digit  $m$ . This is what the function `radix_sort_max` does.

The function `sort_nth_digit` sorts the list by digit  $n$  in linear time by putting the integers whose  $n$ th digit of value  $k$  in the  $k$ th list using the function `bucket`; after that, the integers in the 0th to 9th lists are concatenated together.

If we don't know the max number of digits in a list, then we have to find that using the function `max_digits` in  $O(n + m)$  time, where  $m$  is the max number of digits of the integers in the list. Then we can implement the function `radix_sort`.

## 3 Style requirement

Please use patterns whenever possible (I don't want to see `hd`, `tl`, and `#` functions). Use multiple function bodies to minimize if-then-else branch. Use `let` expressions and inner functions to hide your helper functions.

Submit your solution in a file `hwk4.sml` to the dropbox by next Friday.