# S8.1: Introdution to Purely Functional Data Structures

CSci 2041:

Advanced Programming Principles

University of Minnesota,
Prof. Van Wyk,
Spring 2018

# Exercise #1:Exercise 2.1

Complete Exercise 2.1 from page 11 of Okasaki.
Solve in groups of 2 or 3.

# Exercise 2.1

```
let rec ss = function
| [] -> [ [] ]
| x::xs -> (x::xs) ::  ss xs
```

- Each application to a non-empty list causes another application of the function. So, we will have $O(n)$ function calls.
- Each application uses constant time and space.
- Each creates 2 new nodes - one for each cons cell

# Exercise #2:

Implement `fromList` with type `int list -> Heap`.

First as one left-to-right `fold`.

Then as $\lceil log\ n \rceil$ passes, merging adjacent pairs of heaps.

# Exercise #3:

In pairs, draw rank 4 and rank 5 trees.

# Exercise #4:Exercise 3.9

Write a function `fromOrdList` of type `int list -> Tree` that converts a sorted list with no duplicates into a red-black tree. Your function should run in $O(n)$ time.

Work in pairs or triples.