CSC 225: Spring 2018: Lab 2
Recursions

January 14, 2018

1 Programming Exercise

We will try to implement some recursive methods using the data structure *linked list*. The template "Recursions.java" can be found on connex in the folder named "Lab2". The methods we will implement today are:

1. **Sum(head)** Return the sum of all elements in the linked list starting at the provided node (head).
2. **FindMax(head)** Return the maximum of all elements in the linked list starting at the provided node (head).
3. **JoinLists(h1, h2)** Given two linked lists of integers, create a list consisting of the first list followed by the second list.
4. **ValuePlusIndex(index, head)** Given a starting index and the first node of a linked list, create a new list containing each node's value plus its index.

   For example, with the list [3, 3, 3] and the starting index 0, the result should be [3, 4, 5].

   With the list [3, 4, 5] and the starting index 0, the result should be [3, 5, 7].

   With the list [3, 4, 5] and the starting index 1, the result should be [4, 6, 8].
5. **ExtractEvenNodes(head)** Given a linked list of integers (starting at the provided node), construct a new list containing the even numbers in the input list.

   For example, if the input list is [3, 4, 6, 7, 8, 9, 11], the result should be [4, 6, 8].

2 More exercise ideas

Some basic mathematical functions that you can calculate easily using recursions.

1. **Anagrams.** An anagram is word or phrase formed by rearranging the letters of a different word or phrase. Given a word or phrase, can you compute all the anagrams? They don't have to make sense all the time. For example the anagrams of *sea* is *eas*, *esa*, *sae*, *aes*, and *ase*.

2. **Fibonacci Series.** The series $1, 1, 2, 3, 5, 8, 13, 21, \ldots$ is called the Fibonacci series. The 0th and 1st numbers are 1, after that every number is computed by adding the two numbers appearing right before it. So, the second number is the sum of the zeroth and the first number, the third number is the sum of the first and second number and so on. Quoting from Wikipedia "Fibonacci sequences appear in biological settings such as branching in trees, arrangement of leaves on a stem, the fruitlets of a pineapple, the flowering of artichoke, an uncurling fern and the arrangement of a pine cone, and the family tree of honeybees.". Given an integer $n \geq 0$, Can you compute the nth Fibonacci number?

3. **Factorial.** The factorial of 0 is 1. And the factorial of any number $n > 0$ is the product $1 \times 2 \times \ldots \times n$.

4. **Pascal's triangle.** It is a triangular representation of the binomial coefficients $\binom{n}{k}$, where $n \geq 0, k \geq 0$, as shown in Figure 1. You can compute the entries by making use of this equation:

$$\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}$$

An interesting fact is that if you sum up the numbers of the nth row of

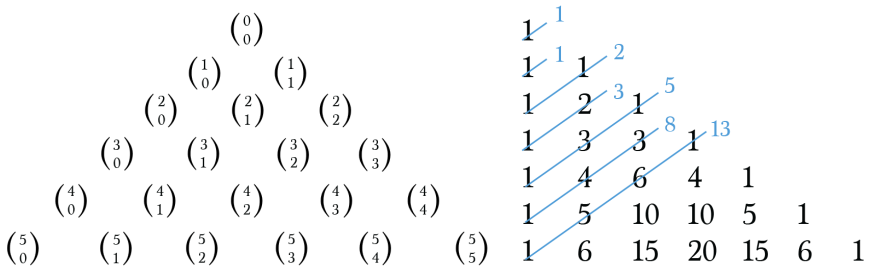

Figure 1: Pascal's triangle (Source: https://medium.com/i-math/top-10-secrets-of-pascals-triangle-6012ba9c5e23)

Pascal's triangle, you get the $(n-1)$th power of 2 or 2^{n-1}, where $n \geq 1$.

On the other hand, if you add the numbers diagonally as in the right figure, you get the Fibonacci numbers. Given the input $n \geq 1$, can you show the first n rows of Pascal's triangle?

You can find some more advanced and more beautiful recursive method ideas here: http://www.toves.org/books/java/ch18-recurex/

1. **Sierpinski Carpet.**
2. **Tree or other structures using Fractal Equations.**