**CPSC449 W23 - Assignment 4**

**Q1** From textbook (Q14.4, P 331)

Using the Expr data type

data Expr = Lit Integer

| Add Expr Expr

| Sub Expr Expr

Define function size which counts the number of operators in an expression.

size :: Expr -> Integer

**Q2** From textbook (Q 14.13, P 332)

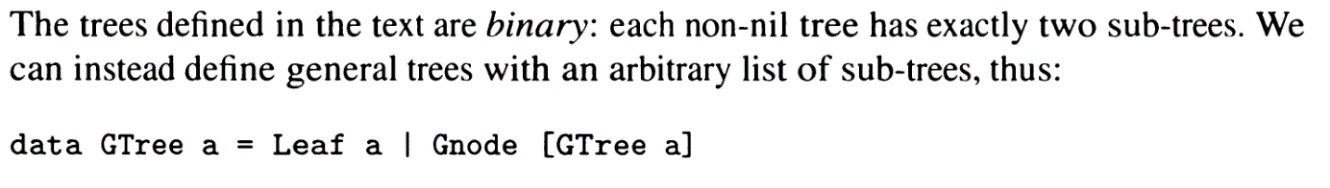
Using the NTree data type:

data NTree = NilT | Node Integer NTree NTree

Define function collapse that enumerates the left subtree then the value at the node and finally the right subtree.

collapse :: NTree -> [Int]

**Q3** From textbook (Q14.21, P 337)



Using this definition, define functions:

1. numberOfLeaves :: GTree a -> Int .Which counts the number of leaves of a GTree
2. depth:: GTree a -> Int . Which returns the depth of a GTree
3. findVal :: a -> GTree a -> Bool. Which searches a value in a GTree

**Q4** From textbook (Q 17.4, P 428)

Given the type Vector, the scalarProduct on two vectors can be defined using list comprehension as:  
scalarProduct xs ys = sum [ x \* y | (x, y) <- zip xs ys ]

Rewrite scalarProduct using zipWith.

**Q5.** From textbook (Q17.23, P 446)

Give a definition of the infinite list of factorials

factorial = [1, 1, 2, 6, 24, 120, 720, …]

Please submit all your codes and screenshots of the running output in one PDF or Word document on D2L.