Sujeeth Shetty 2/2/2021

BUAN 6347 Advance Big Data

Homework 1

Question 1:

a) The GoodEnough test takes the absolute difference. The value given as 0.001 for the boundary is imprecise for small numbers. In case of very large floating point the difference might be larger than the 0.001.

b) By dividing by x regardless of the scale of x the difference should maintain a constant percentage via its fractional value

```
def isGoodEnough(guess: Double, x: Double) =
  abs(guess * guess - x)/x < 0.01
```

Question 2:

a)

```
def union(other: IntSet ): IntSet =((left union right) union other) incl elem

def intersection (other: IntSet): IntSet ={
  val newSet = (right intersection other) union (left intersection other)
  if (other contains elem) newSet incl elem
  else newSet
}

def isEmpty = false

def excl(x: Int): IntSet = {
  if (elem == x) left union right
  else if (elem < x) (left excl x) union right incl elem
  else left union (right excl x) incl elem
```

Question 3:

a)

```
case class Integer(value: Nat, sign : Sign = Positive) extends Nat with Sign {

  def isZero : Boolean = value.isZero

  def predecessor: Nat = {
    if (isZero) new Integer(value.successor, Negative)
    else if (sign.isPositive) new Integer(value.predecessor, sign)
    else new Integer (value.successor, Negative)
  }

  def successor: Nat = {
    if (isZero) new Integer(value.successor, Positive)
    else if (sign.isPositive) new Integer (value.successor, sign)
    else new Integer (value.successor, Negative)
  }
```

```
def +(that: Nat): Nat = {
  if (isZero) that
  else if (sign.isPositive) this.predecessor + that.successor
  else this.successor + that.predecessor
}

def -(that:Nat): Nat=
  if(that.isZero) this
  else that match {
    case Integer(v,s) => this + new Integer(v, s.negate)
  }

def isPositive: Boolean = sign.isPositive

def negate: Integer = new Integer(value, sign.negate)

val toInt: Int = if (sign.isPositive) value.toInt else -value.toInt
```

Reference:

Coursera: Functional Programming Principles in Scala