Recitation Session Solutions, October 11th, 2017

Ex 3

```
def toList: List[T] = {
  def toListAcc(tree: Tree[T], acc: List[T]): List[T] = tree match {
    case EmptyTree(_) => acc
    case Node(left, elem, right, _) =>
      toListAcc(left, elem :: toListAcc(right, acc))
  }

  toListAcc(this, Nil)
}
```

Ex 4

```
def sortedList[T](leq: (T, T) => Boolean, ls: List[T]): List[T] = {
  def buildTree(elems: List[T], acc: Tree[T]): Tree[T] = elems match {
    case Nil => acc
    case elem :: rest => buildTree(rest, acc.add(elem))
  }

  buildTree(ls, EmptyTree(leq)).toList
}
```