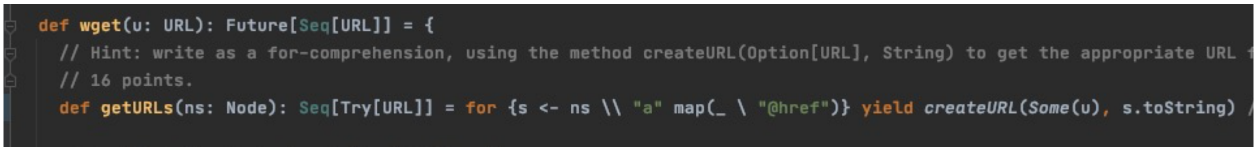
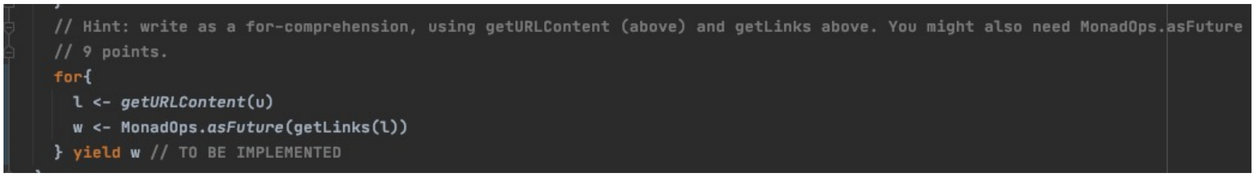
Jingru Xiang – (NUID 001586653)

Big Data System Engineering with Scala
Spring 2022
Assignment No. 6

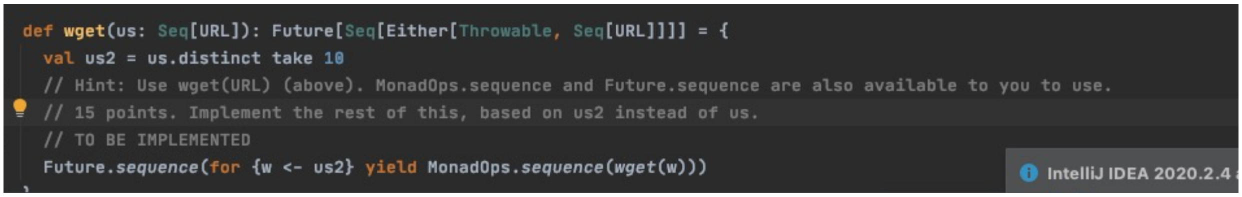
Code:



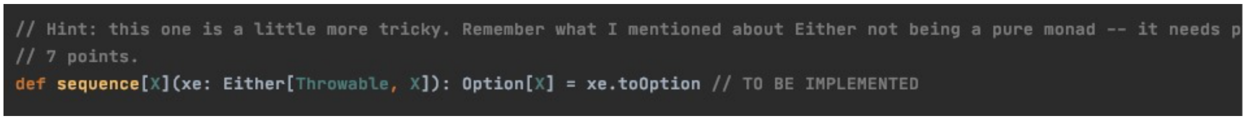
```
def wget(u: URL): Future[Seq[URL]] = {
  // Hint: write as a for-comprehension, using the method createURL(Option[URL], String) to get the appropriate URL f
  // 16 points.
  def getURLs(ns: Node): Seq[Try[URL]] = for {s <- ns \\ "a" map(_ \ "@href")} yield createURL(Some(u), s.toString) /
```



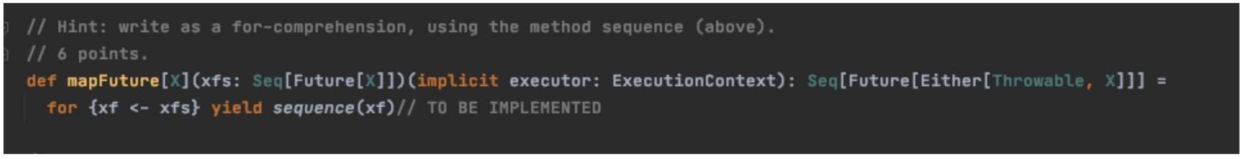
```
  // Hint: write as a for-comprehension, using getURLContent (above) and getLinks above. You might also need MonadOps.asFuture
  // 9 points.
  for{
    l <- getURLContent(u)
    w <- MonadOps.asFuture(getLinks(l))
  } yield w // TO BE IMPLEMENTED
```



```
def wget(us: Seq[URL]): Future[Seq[Either[Throwable, Seq[URL]]]] = {
  val us2 = us.distinct take 10
  // Hint: Use wget(URL) (above). MonadOps.sequence and Future.sequence are also available to you to use.
  // 15 points. Implement the rest of this, based on us2 instead of us.
  // TO BE IMPLEMENTED
  Future.sequence(for {w <- us2} yield MonadOps.sequence(wget(w)))
```

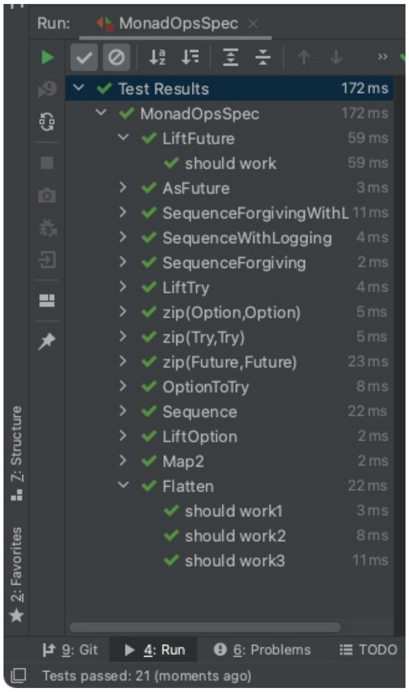


```
// Hint: this one is a little more tricky. Remember what I mentioned about Either not being a pure monad -- it needs p
// 7 points.
def sequence[X](xe: Either[Throwable, X]): Option[X] = xe.toOption // TO BE IMPLEMENTED
```

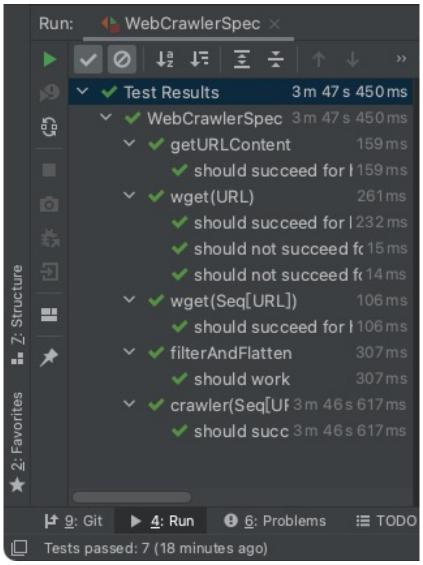


```
// Hint: write as a for-comprehension, using the method sequence (above).
// 6 points.
def mapFuture[X](xfs: Seq[Future[X]])(implicit executor: ExecutionContext): Seq[Future[Either[Throwable, X]]] =
  for {xf <- xfs} yield sequence(xf)// TO BE IMPLEMENTED
```

Unit Test:



```
Test Results                 172 ms
  MonadOpsSpec               172 ms
    LiftFuture                59 ms
      should work             59 ms
    AsFuture                   3 ms
    SequenceForgivingWithL    11 ms
    SequenceWithLogging        4 ms
    SequenceForgiving          2 ms
    LiftTry                    4 ms
    zip(Option,Option)         5 ms
    zip(Try,Try)               5 ms
    zip(Future,Future)        23 ms
    OptionToTry                8 ms
    Sequence                  22 ms
    LiftOption                 2 ms
    Map2                       2 ms
    Flatten                   22 ms
      should work1             3 ms
      should work2             8 ms
      should work3            11 ms
Tests passed: 21 (moments ago)
```



```
Test Results                      3 m 47 s 450 ms
  WebCrawlerSpec                  3 m 47 s 450 ms
    getURLContent                          159 ms
      should succeed for l                 159 ms
    wget(URL)                              261 ms
      should succeed for l                 232 ms
      should not succeed fc                 15 ms
      should not succeed fc                 14 ms
    wget(Seq[URL])                         106 ms
      should succeed for l                 106 ms
    filterAndFlatten                       307 ms
      should work                          307 ms
    crawler(Seq[UI                3 m 46 s 617 ms
      should succ                 3 m 46 s 617 ms
Tests passed: 7 (18 minutes ago)
```

How to improve:

Using Akka stream with its asynchronous and multi-threaded messaging model to utilize hardware.

Project Source: https://github.com/RubyXiang/CSYE7200/tree/Spring2022/assignment-web-crawler/src