Quiz: Algebraic Data Types

9 minutes, no stress, no embarrassment, no consequences, but **alone and quietly**

- Write an ADT in Scala for representing binary numbers of arbitrary size.
- A binary is either zero,
- or it is one,
- or it is a digit (one, zero) followed by a shorter binary number

Quiz: Algebraic Data Types

9 minutes, no stress, no embarrassment, no consequences, but **alone and quietly**

- Write an ADT in Scala for representing binary numbers of arbitrary size.
- A binary is either zero,
- or it is one,
- or it is a digit (one, zero) followed by a shorter binary number

```scala
sealed trait Binary
case object One extends Binary
case object Zero extends Binary
case class O (t: Binary) extends Binary
case class I (t: Binary) extends Binary
```

Quiz: Algebraic Data Types

9 minutes, no stress, no embarrassment, no consequences, but **alone and quietly**

- Write an ADT in Scala for representing binary numbers of arbitrary size.
- A binary is either zero,
- or it is one,
- or it is a digit (one, zero) followed by a shorter binary number

```
1 sealed trait Binary
2 case object One extends Binary
3 case object Zero extends Binary
4 case class O (t: Binary) extends Binary
5 case class I (t: Binary) extends Binary
```

```
1 sealed trait Binary
2 case object Empty extends Binary
3 case class O (t: Binary) extends Binary
4 case class I (t: Binary) extends Binary
```

Quiz: Algebraic Data Types

9 minutes, no stress, no embarrassment, no consequences, but **alone and quietly**

- Write an ADT in Scala for representing binary numbers of arbitrary size.
- A binary is either zero,
- or it is one,
- or it is a digit (one, zero) followed by a shorter binary number

```
1 sealed trait Binary
2 case object One extends Binary
3 case object Zero extends Binary
4 case class O (t: Binary) extends Binary
5 case class I (t: Binary) extends Binary
```

```
1 sealed trait Binary
2 case object Empty extends Binary
3 case class O (t: Binary) extends Binary
4 case class I (t: Binary) extends Binary
```

- Try to represent 101, 0, 1: `I(O(One))` for the left, `I(O(I(Empty)))` for the right
- Success? $\rightarrow$ award 4 points.
- Minor problems? $\rightarrow$ subtract a point. Don't penalize for very minor issues.
- Zero points if the solution is polymorphic, or is not recursive.