Lesson 6

Algorithms: The Right-Hand Rule

```
func navigateAroundWall()
{
    if isBlockedRight {
        moveForward()
    } else {
        turnRight()
        moveForward()
    }
}
while !isOnOpenSwitch() {
    while !isOnGem II !isOnClosedSwitch {
        while !isBlocked {
            navigateAroundWall()
        }
        if isOnGem {
            collectGem()
        } else {
            toggleSwitch()
        }
        turnLeft()
        turnLeft()
}
```

Algorithms: Adjusting Your Algorithm

```
func navigateAroundWall() {
    if isBlockedRight && isBlocked {
        turnLeft()
    }  else if isBlockedRight {
        moveForward()
    } else {
        turnRight()
        moveForward()
    }
    if isOnGem {
        collectGem()
    }
}

while !isOnClosedSwitch {
    navigateAroundWall()
}
toggleSwitch()
```

Lesson 6

Algorithms: Conquering a Maze

```
func navigateAroundWalls() {
    if isBlockedRight && isBlocked {
        turnLeft()
    }  else if isBlockedRight {
        moveForward()
    } else {
        turnRight()
        moveForward()
    }
}

while !isOnGem {
    navigateAroundWalls()
}
collectGem()
```

*Some puzzles may have multiple solutions

Lesson 6

Algorithms: Conquering a Maze

```
func navigateAroundWalls() {
    if isBlockedRight && isBlocked {
        turnLeft()
    }  else if isBlockedRight {
        moveForward()
    } else {
        turnRight()
        moveForward()
    }
}

while !isOnGem {
    navigateAroundWalls()
}
collectGem()
```

*Some puzzles may have multiple solutions