Activity Diagram
Paul-Emmanuel Courtines (pyc25)

There are two parts to this project; I've outlined each below in their own part. In the first part, **BetterSlot.java**, I detail my solution to the slot machine class. In the latter portion, I present my JUnit test class, **TestSlipperySlot.java**.

BetterSlot.java

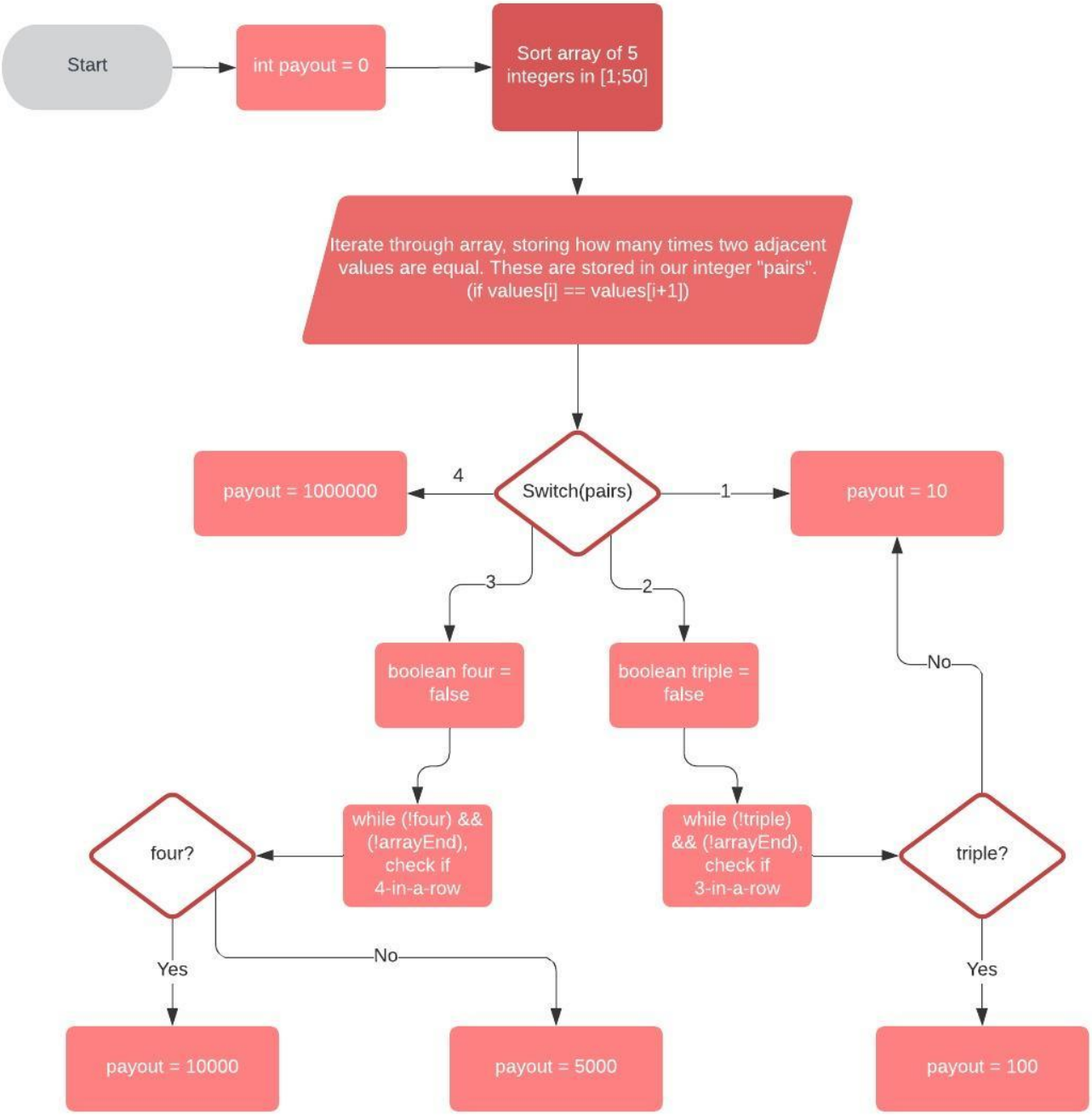
public int[] pullTheLever()

- Generate 5 random numbers between 1 and 50. Assign them to an integer array of size 5 and return.

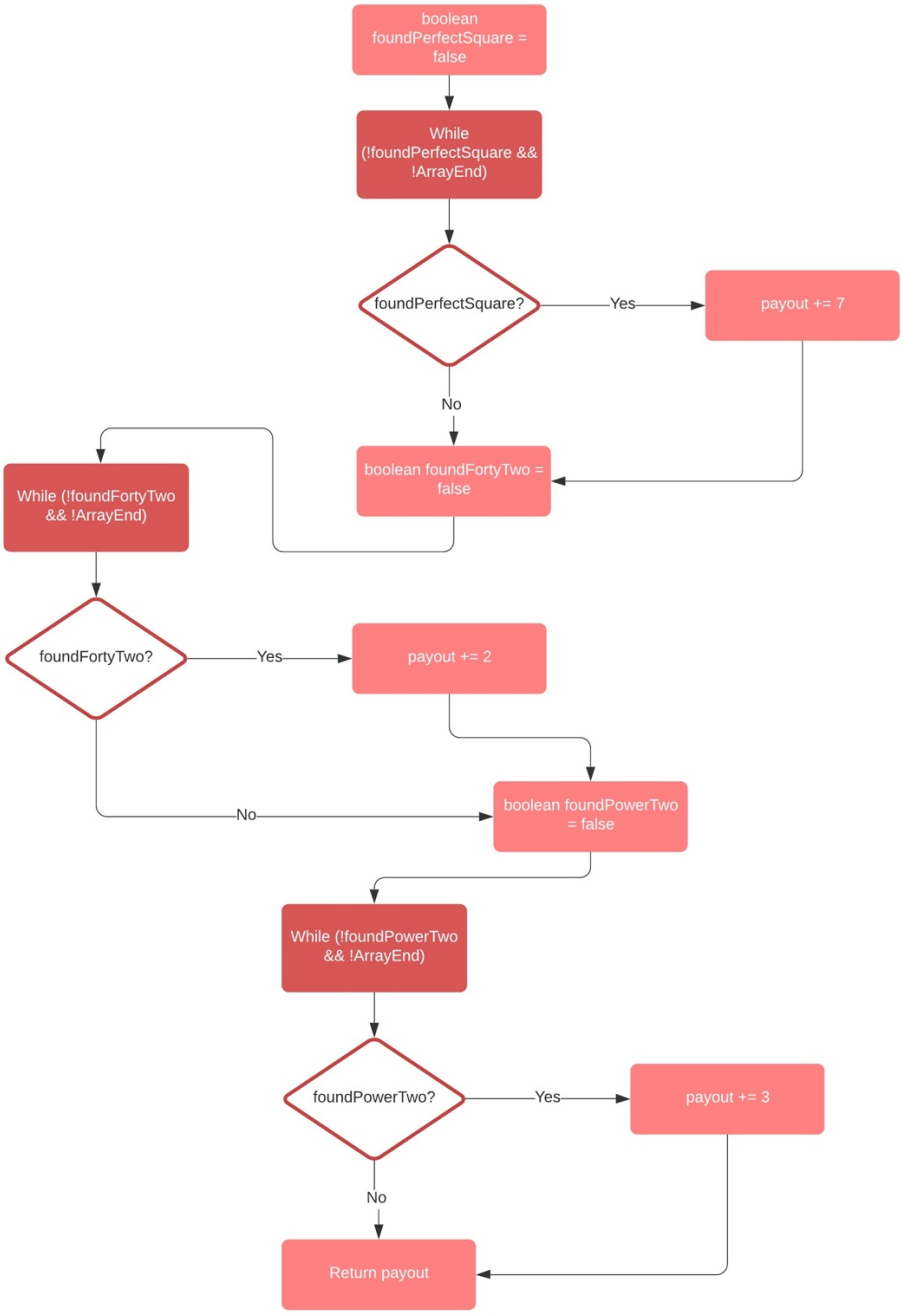
public int payOff(int[] values)

- Given an array of 5 random integers between 1 and 50, begin by ordering the values.

- Iterate through the array and count how many pairs, two adjacent equal values, there are.
- Use a switch method to determine payoff based on number of pairs.
    - If 1: There is only a single pair. Set payout to 10.
    - If 2: Iterate through array, checking if 3 integers in a row are equal. If true, set payout to 100. Else, set payout to 10.
    - If 3: Iterate through array, checking if 4 integers in a row are equal. If true, set payout to 10000. Else, set payout to 5000.
    - If 4: There is a full house. Set payout to 1000000.



- To test cases 6, 7 and 8, we use three while loops, iterating through the arrays and testing each value for cases
    - Case 6: Inside the while loop, the if statement checks if sqrt(values[i]) is an integer, in which case it is a perfect square.
    - Case 7: Inside the while loop, the if statement checks if values[i] is equal to 42.
    - Case 8: Inside the while loop, the if statement checks if values[i] is a power of 2.



TestSlipperySlot.java

Test	Values	Description
testLeverPull	NA	Makes 20 calls to SlipperySlot's pullTheLever(). Tests if an array of 5 values between 1 and 50 is returned.
testAllSame	Integer in [1;50]	Creates 20 array of 5 identical values between 1 and 50. Checks that return is greater or equal to 1000000.
testSimplePairOne	5, 5, 7, 11, 43	Tests return of 10.
testSimplePairTwo	3, 3, 39, 37, 14	Tests return of 10.
testTwoPairsOne	3, 3, 39, 39, 14	Tests return of 10.
testTwoPairsTwo	5, 5, 14, 14, 23	Tests return of 10.
testTripleOne	39, 39, 14, 39, 41	Tests return of 100.
testTripleTwo	3, 3, 3, 37, 14	Tests return of 100.
testFullHouseOne	3, 3, 37, 37, 3	Tests return of 5000.
testFullHouseTwo	23, 7, 23, 23, 7	Tests return of 5000.
testFourOne	3, 3, 37, 3, 3	Tests return of 10000.
testFourTwo	23, 7, 23, 23, 23	Tests return of 10000.
testFortyTwoOne	3, 5, 6, 7, 42	Tests return of 2.
testFortyTwoTwo	40, 41, 42, 43, 44	Tests return of 2.
testFortyTwoAndPair	3, 5, 7, 42, 42	Tests return of 12.
testAllFortyTwo	42, 42, 42, 42, 42	Tests return of 1000002.

testPerfectSquareOne	40, 41, 43, 44, 49	Tests return of 7.
testPerfectSquareTwo	3, 9, 27, 36, 37	Tests return of 7.
testPerfectSquareAndPair	9,9,11,13,14	Tests return of 17.
testAllPerfectSquare	49, 49, 49, 49, 49	Tests return of 1000007.
testPowerTwoOne	3, 8, 11, 32, 33	Tests return of 3.
testPowerTwoTwo	30, 31, 32, 33, 34	Tests return of 3.
testAllPowerTwo	8, 8, 8, 8, 8	Tests return of 1000003.
testAllPerfectSquareAndPowerTwo	4, 4, 4, 4, 4	Tests return of 1000010.
testFortyTwoAndPerfectSquareAndPowerTwo	1, 4, 16, 42, 49	Tests return of 12.
testTimeout	NA	Makes 10,000 calls to SlipperySlot's pullTheLever() and to payOff(). Set timeout somewhere in [40;45], so as to allow correct implementation to pass and fail incorrect ones.