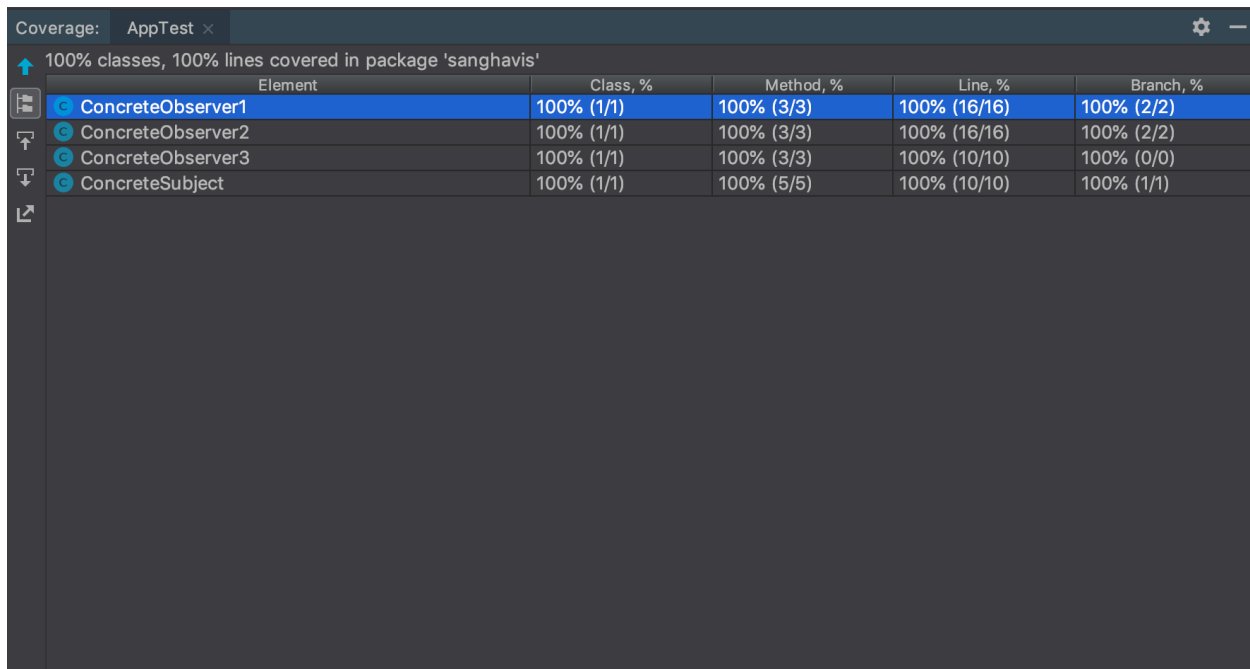1) **Q1: TestSuite Justification**
   The test suit consists of 5 tests which achieves 100% Branch coverage and 100% Condition coverage.

   From function point of view, the edge cases such as empty string, null string and a random string is provided to the state of the subject. Each attached observer were expected to return appropriate output.

   In first test, subject state is set to a string and all three observers are expected to update when subject.state is updated. In second test, two observers are detached before calling setState() method this should make only one observer print the output. In third test, all the observer should notify even if an empty string is passed to the setState. It should raise NullPointerException if one tries to set state=null.

   Thus, the test suit checks for general cases and edge cases.

Coverage:	AppTest ×				⚙ —
100% classes, 100% lines covered in package 'sanghavis'					
Element		Class, %	Method, %	Line, %	Branch, %
ConcreteObserver1		100% (1/1)	100% (3/3)	100% (16/16)	100% (2/2)
ConcreteObserver2		100% (1/1)	100% (3/3)	100% (16/16)	100% (2/2)
ConcreteObserver3		100% (1/1)	100% (3/3)	100% (10/10)	100% (0/0)
ConcreteSubject		100% (1/1)	100% (5/5)	100% (10/10)	100% (1/1)

2) Q2: TestSuite Justification

The test suite consists of 17 test cases which provides 100% branch coverage and 100% condition coverage.

From functional point of view, the test suite checks following methods: constructor method, get, set, clear, copy, size, iterator.

For constructor, it checks whether a BitVector with all unset bits are created or not.

For get method, it checks for the set bit, unset bit, large index of set bit and a large index of unset bit.

For set method, it checks whether program can set a bit for small number, large number, zero and multiple numbers.
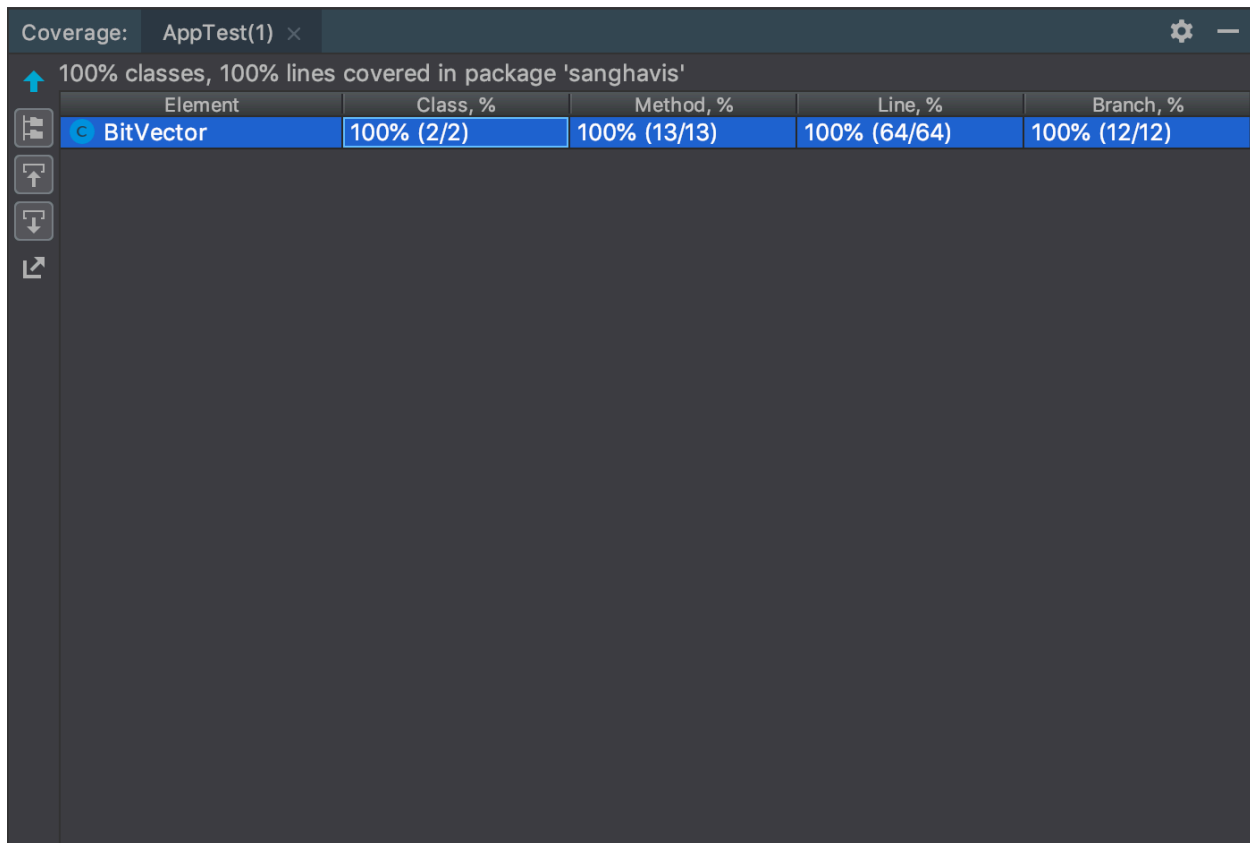
For clear, it checks whether it can clear set bit, unset bit, very large set bit and very large unset bit.

For size, it checks for the size of a bitvector for a set of natural numbers including small numbers and very large numbers. It also checks for the size of bit vector for an empty set.

For copy, it checks, CASE1) whether the small bitvector can be copied to a large bitvector or not and CASE2) whether the large bitvector can be copied to a small bitvector or not.

For iterator, it checks whether iterator method returns an appropriate iterator to iterate over the integers of the bitvector.

Thus, the test suit checks for general cases and edge cases.
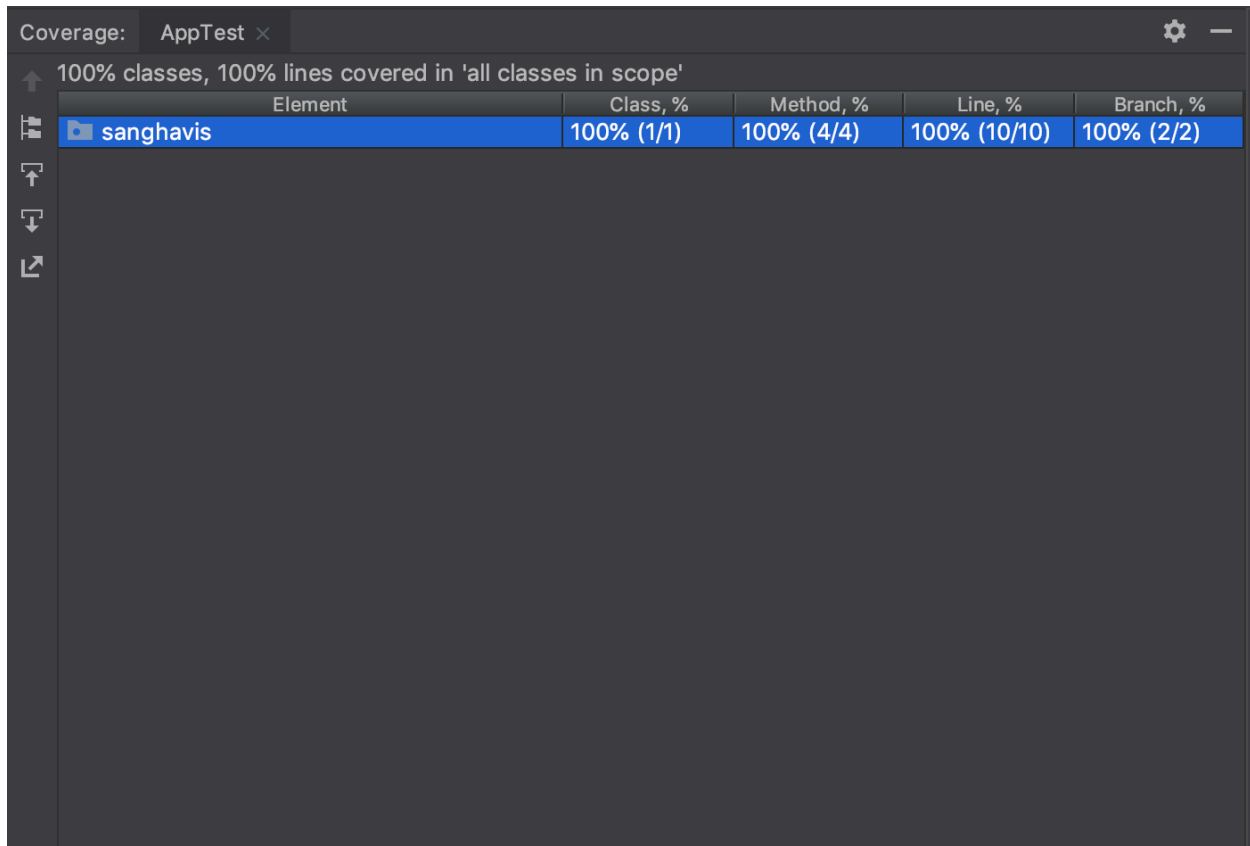
Coverage:	AppTest(1) ×				⚙ —
↑ 100% classes, 100% lines covered in package 'sanghavis'					
Element	Class, %	Method, %	Line, %	Branch, %	
© BitVector	100% (2/2)	100% (13/13)	100% (64/64)	100% (12/12)	

3) Q3: Adapt Queue
The testsuite provides 7 tests which achieves 100% branch coverage and 100% condition coverage.
As the queue is adapted from ArrayList<T>, it should allow to create queue for any data type. It checks whether we can add integers, double, string to a queue or not. It checks the output of remove and peek method when there are multiple objects in the queue. It also checks the output of remove and peek when there is no data in the queue.
Thus, the test suit checks for general cases and edge cases.

Coverage:	AppTest ×				
100% classes, 100% lines covered in 'all classes in scope'					
Element		Class, %	Method, %	Line, %	Branch, %
sanghavis		100% (1/1)	100% (4/4)	100% (10/10)	100% (2/2)