

1. automatically inserts values in the correct position based on some order of sorting (perhaps ascending integers or lexicographical sorting of words)

```
10 public:
11     void insert(int value) {
12         auto it = std::lower_bound(array.begin(), array.end(), value);
13         array.insert(it, value);
14     }
15
```

2. efficiently searches for elements (likely binary search for the array list, but what about the linked-list?)

```
bool search(int value) {
    return std::binary_search(array.begin(), array.end(), value);
}

};
```

Testing:

Testing Insert function:

```
21 void testArrayList() {
22     SortedArrayList list;
23
24     // Test insert function in random order
25     list.insert(3);
26     list.insert(2);
27     list.insert(4);
28     list.insert(1);
29     std::cout << "Test 1: " << (list.search(4) ? "Passed" : "Failed") << std::endl;
30
```

Testing search function:

```
30
31     // Test search function
32     assert(list.search(1));
33     assert(list.search(2));
34     assert(list.search(3));
35     assert(list.search(4));
36
37     std::cout << "Search function tests passed successfully. " << std::endl;
38
```