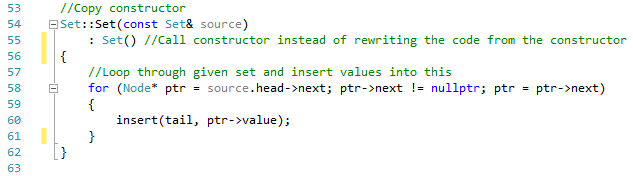
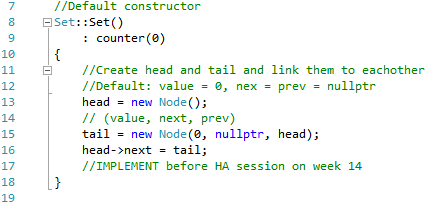
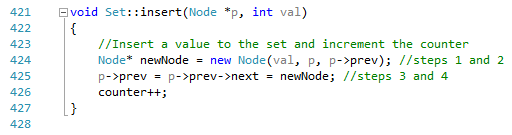
Lab 2 exercise 3

1. Set S3(S1);

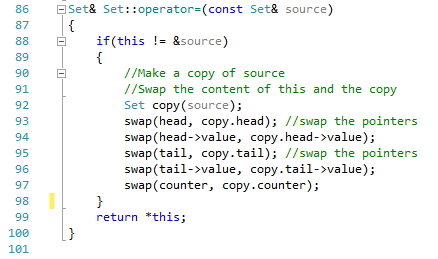






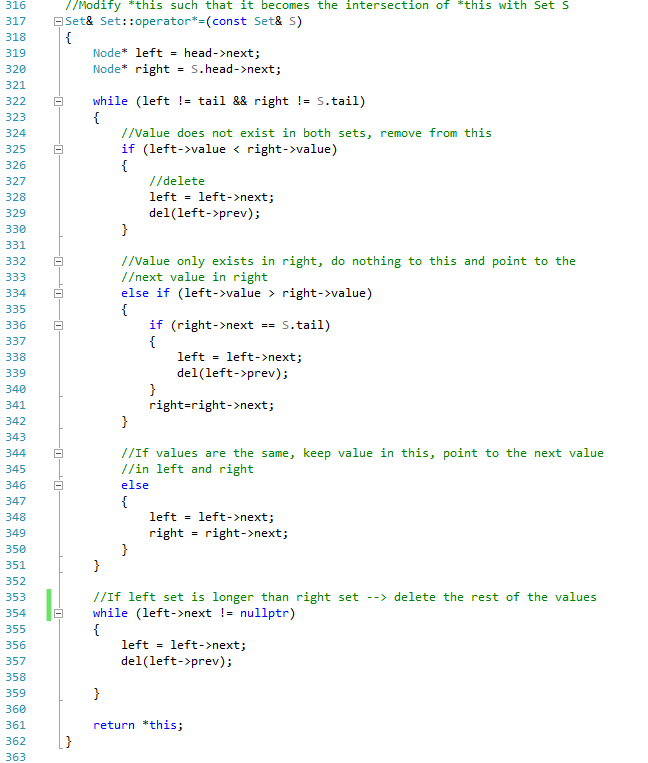
This line of code calls the default constructor and the copy constructor, which calls the insert function. As seen in the default constructor, the execution takes constant time (C1). The for-loop in the copy constructor loops through the number of values (n) in the Set and calls the insert function for each value in the set. The insert function is also executed in constant time (C2). Therefore, the total time complexity is:

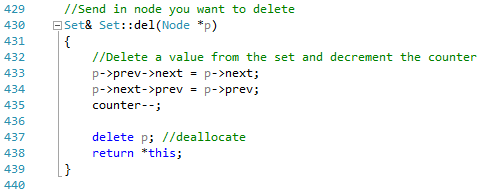
1. S1 = S2;



This line of code calls the assignment operator. In the operator a comparison is being made and then the copy constructor is called which is of time complexity O(n). The swap function is of constant time (C). Therefore, the total time complexity is:

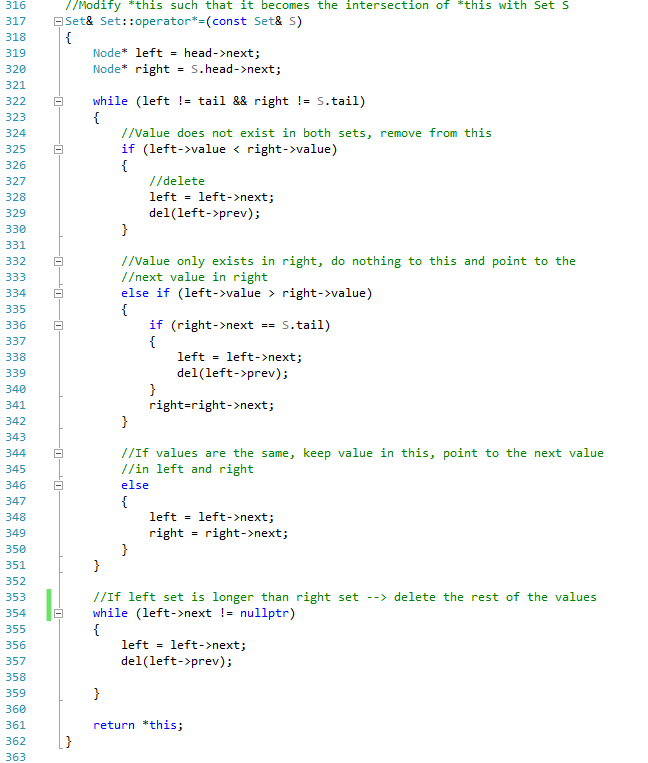
1. S1 \* S2;





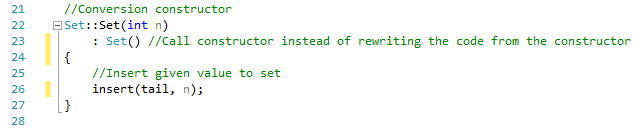
This line of code calls operator\* which calls the intersection function. We will calculate the time complexity for the worst case scenario. The worst case scenario is that S1 contains more values than S2. First, two nodes are assigned in constant time (C1). Then the first while-loop loops through the number of values in S2 (m). The delete function is called inside two of the if-statements. The delete function is executed in constant time. Therefore, the code in the first while-loop is also executed in constant time (C2). The second while-loop will loop through the remaining values of S1 (n-m) where S1 contains n values. The code in the second while-loop is also executed in constant time (C3). This will result in the time complexity:

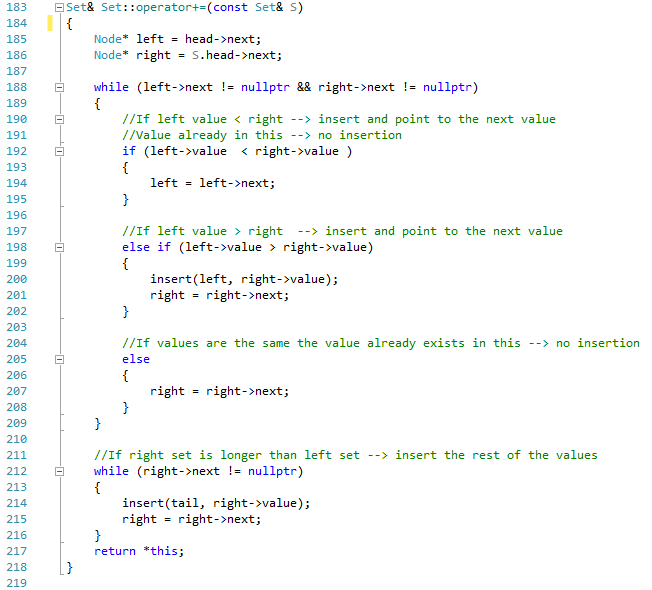
1. std::move(S1) \* S2;



This line of code “marks” S1 to be a rvalue reference. Then the operator\* is called which calls operator \*=. As in task 3, we can see that this results in:

1. k + S1;





This line of code calls the conversion constructor to convert the constant into a Set. The conversion constructor calls the default constructor and the insert function which both operates in constant time as said before. Therefore, this results in constant time. After this, the union function is called. We will calculate the time complexity for the worst case scenario. The worst case scenario is that S2 contains more values than S1. First, two nodes are assigned in constant time (C1). Then the first while-loop loops through the number of values in S1 (m). The insert function is called inside one of the if-statements and is executed in constant time. Therefore, the code in the first while-loop is also executed in constant time (C2). The second while-loop will loop through the remaining values of S2 (n-m) where S2 contains n values. The code in the second while-loop is also executed in constant time (C3). This will result in the time complexity: