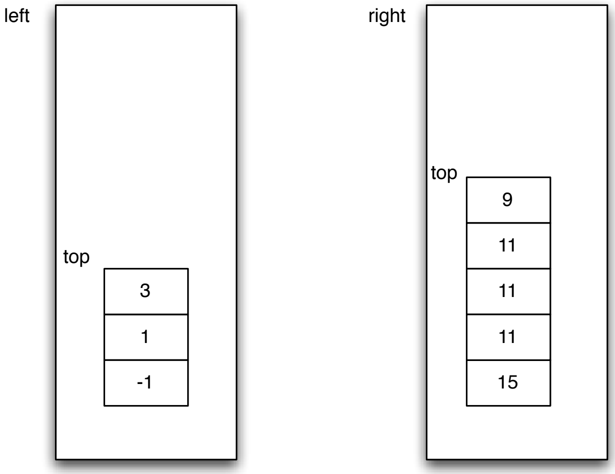
Program 5.1 The StackSort Class

Visualization

Stack Sort

In order to sort values, we will use two stacks which will be called the left and right stacks. The values in the stacks will be sorted and the values in the left stack will all be less than or equal to the values in the right stack. The following example illustrates a possible state for our two stacks. Notice that the values in the left stack are sorted so that the smallest value is at the bottom of the stack. The values in the right stack are sorted so that the smallest value is at the top of the stack. If we read the values up the left stack and then down the right stack, we get -1, 1, 3, 9, 11, 11, 11, 15, which is in sorted order.



Suppose that we have a new value that we want to put into our sorted collection. We will want to put it on the top of one of the two stacks, but we may have to first move values around.

No moves required:

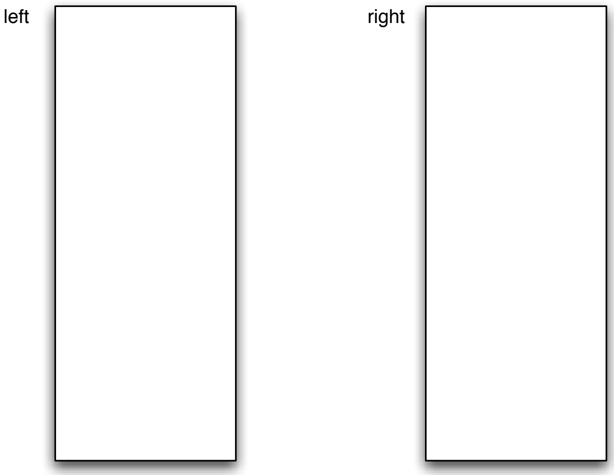
Consider adding the value 5 to the example shown above. We do not have to move any values and can place the 5 on the top of either stack and still have a sorted collection.

Which values would not require that the contents of the stacks be changed?

Moves from left to right required:

Consider adding the value 0 to the example shown above. We must move values from the left stack to the right stack.

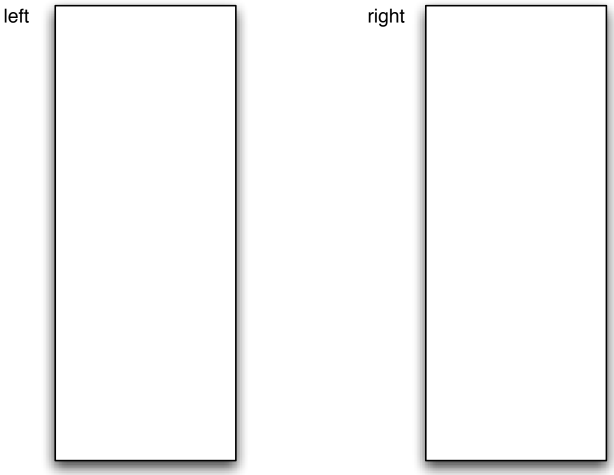
How many values must be moved and what is the state of the two stacks before we add the value 0?



What condition should we use to determine if enough values have been moved?

Consider adding the value -2 to the example shown above. Again must move values from the left stack to the right stack.

How many values must be moved and what is the state of the two stacks before we add the value -2?



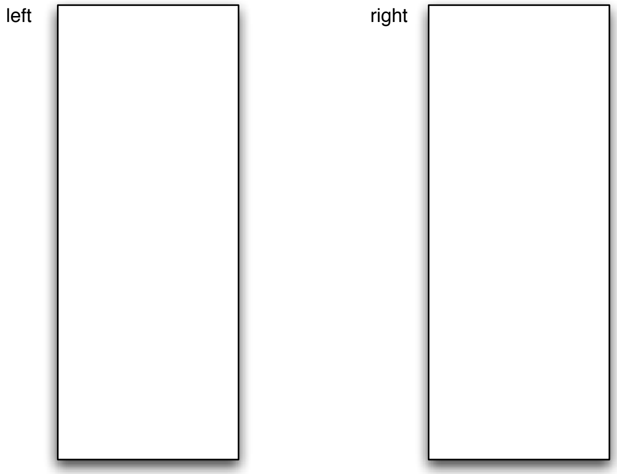
What condition should we use to determine if enough values have been moved?

Write pseudocode using iteration that will move values from the left to the right stack as required.

Moves from right to left required:

Consider adding the value 11 to the example shown above. We must move values from the right stack to the left stack.

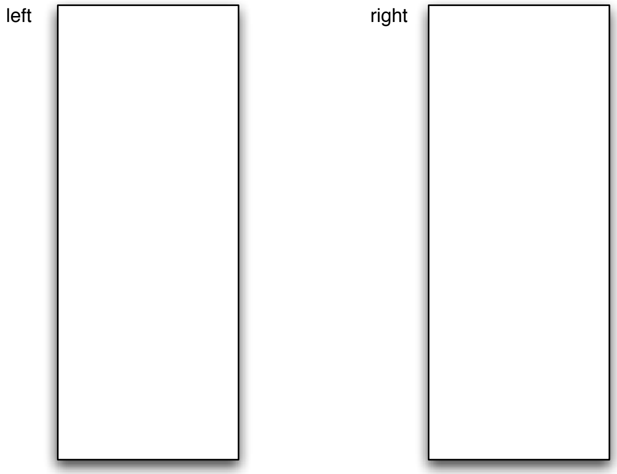
How many values must be moved and what is the state of the two stacks before we add the value 11?



What condition should we use to determine if enough values have been moved?

Consider adding the value 20 to the example shown above. Again we must move values from the right stack to the left stack.

How many values must be moved and what is the state of the two stacks before we add the value 20?

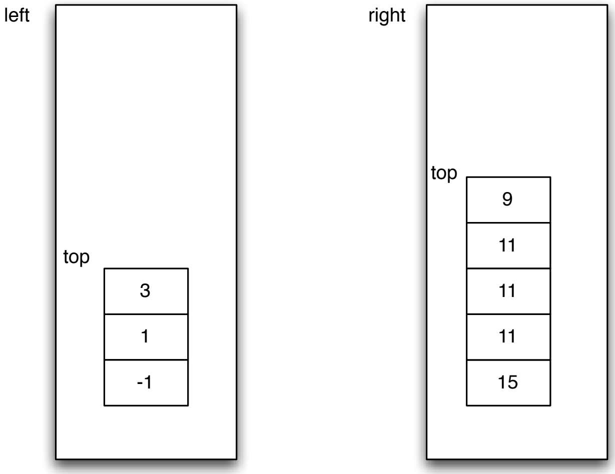


What condition should we use to determine if enough values have been moved?

Write pseudocode using iteration that will move values from the right to the left stack as required.

Putting the values into a new array to return to user:

For our particular sorting algorithm, we are going to create a second array with the values from the original array in sorted order. Therefore, the final task we need to do before we return is to put the values into the `result` array. Consider again our example.



Suppose we pop the values off of the left stack one at a time and push them on right stack, then pop all the values off the right stack. What order do we get?

Suppose we pop the values off of the right stack one at a time and push them on left stack, then pop all the values off the left stack. What order do we get?

This suggests that if we move the values from the one stack to the other stack, we can then directly pop them off of the stack into the `result` array in a given order. Write an algorithm that accomplishes this task.