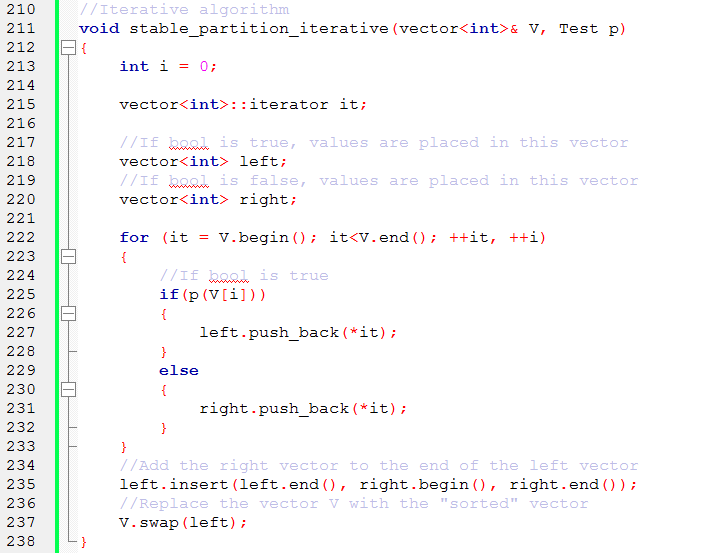
Lab 1 Exercise 3

# **Analysis of iterative algorithm**



## Time Complexity

We have counted everything (initializations, test, increments and modifiers) . For this algorithm there is no “worst case”, regardless if the Boolean is true or false the algorithm will still have the same time complexity.

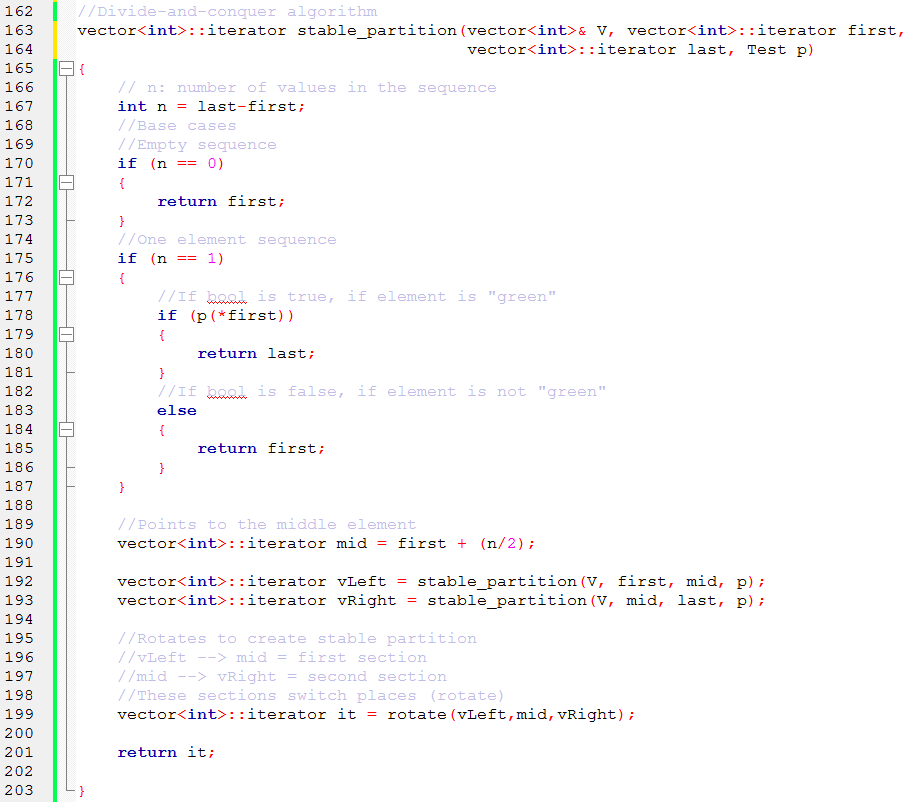
|  |  |  |
| --- | --- | --- |
| Line | Time unit | Comment |
| 213 | 1 | Initialization of int |
| 222 | 1+(n+1)+2\*n | Initialization of iterator, test condition, increment two variables |
| 225-232 | n+ | n: initialization of int in bool function  c: constant complexity for push\_back |
| 235 | n | Insert is linear to the number of elements inserted |
| 237 |  | Swap is of constant complexity |
|  | | |

## Space Complexity

We have counted everything that needs space in the stack except for the Boolean as input. This is for the “worst case” where all values in the sequence are false. If all values are false, vector right will also be of space n.

|  |  |  |
| --- | --- | --- |
| Line | Space unit | Comment |
| 211 | n | Space for sequence V |
| 213 | 1 | Space for integer |
| 215 | 1 | Space for iterator |
| 218 | n | Space to hold sorted V |
| 220 | n | Space to hold values if bool is false. |
|  | | |

# **Analysis of Divide-and-conquer algorithm**



# Time Complexity

We have only counted function calls. There is no “worst case” since the vector will constantly be divided in to two vectors regardless of the input if n>1.

## Applying Master theorem:

Constants:

Since we get:

# Space Complexity

We count the space for the sequence V and the space needed in the function. For n<2 we will only need space for int n(line 167). The are two recursive calls but there will only be one activation record on stack at a time.

## Applying Master theorem:

Constants:

Since we get:

For sequence V, we need space n, therefore we need to add n to S(n), therefore

# Results

|  |  |  |
| --- | --- | --- |
|  | Iterative algorithm | Divide & conquer algorithm |
| Time complexity |  |  |
| Space complecity |  |  |

The iterative algorithm is preferred since the time complexity has a slower growth rate. As seen in the calculations, the iterative algorithm also requires less space since it does not have a logarithmic component.