C++ Assignment 6

April 18, 2023

yijan

Implementations

In a `LinkedList` we have some nodes and there's a double value in each node.

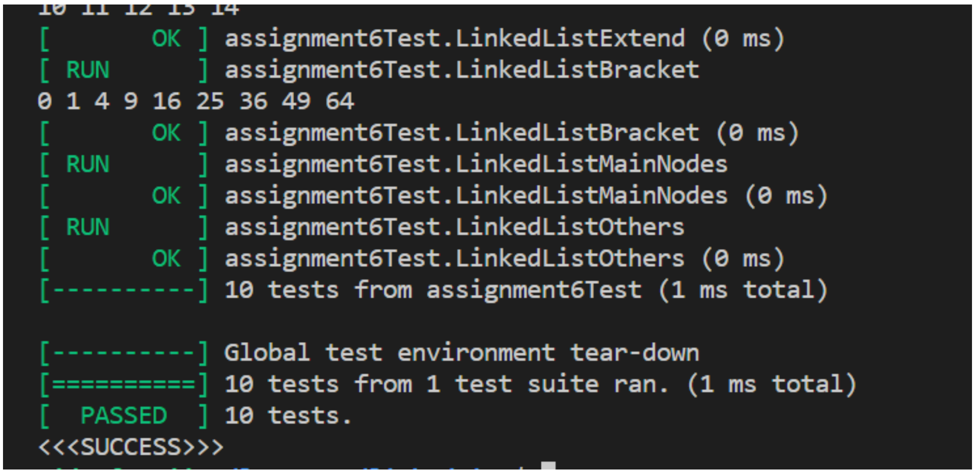
Every nodes is linked with the next node and previous node except the first and last node so that we can visit every node from anyone in the list. I defined a null node on the tail to indicate the end of the list and that node has `nullptr` as next node. The value of null node is `0`.

If the list is empty, `head` and `tail` point to same node (the last node). To get the size in $O(1)$ time, a private value N is declared indicating the current size.

We can maintain the list when pushing on the back or front in O(1) time by changing the pointers.

It's important that the copy constructor needs to copy all nodes but not just copy the head pointer. Copy constructor also needs to assign enough memory to store the value. It's also easy to implement move constructor by moving the pointer.

Result



```
10 11 12 13 14
[       OK ] assignment6Test.LinkedListExtend (0 ms)
[ RUN      ] assignment6Test.LinkedListBracket
0 1 4 9 16 25 36 49 64
[       OK ] assignment6Test.LinkedListBracket (0 ms)
[ RUN      ] assignment6Test.LinkedListMainNodes
[       OK ] assignment6Test.LinkedListMainNodes (0 ms)
[ RUN      ] assignment6Test.LinkedListOthers
[       OK ] assignment6Test.LinkedListOthers (0 ms)
[----------] 10 tests from assignment6Test (1 ms total)

[----------] Global test environment tear-down
[==========] 10 tests from 1 test suite ran. (1 ms total)
[  PASSED  ] 10 tests.
<<<SUCCESS>>>
```