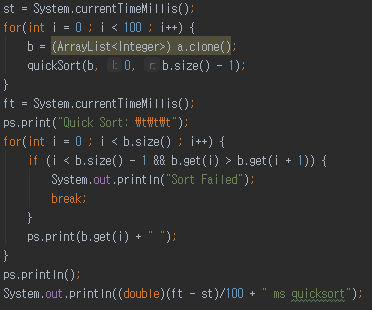
**Data Structure Assignment 4 Result Description**

**2017-12751 Donghak Lee**

We implemented 7 sorting algorithms and tested with random input data. To improve the accuracy, repeated each sorting for 100 times and get average time consumed. Also write a code to verify the output produced by a sorting algorithm is correct.

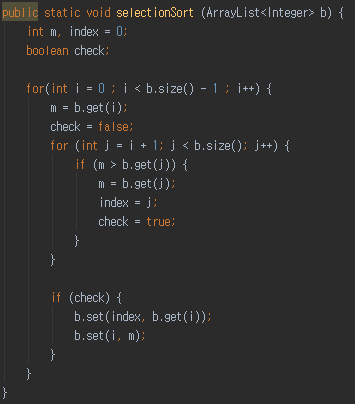
0. Time performance, repeat & sort checking

|  |  |
| --- | --- |
| Result of 100 inputs | Result of 1000 inputs |
| Result of 10000 inputs | Result of 100000 inputs |



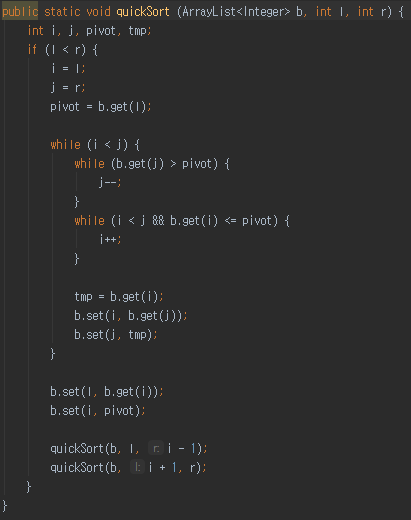
Repeat each sort 100 times and if algorithm is wrong, print “Sort Failed” and break.

1. Selection sort



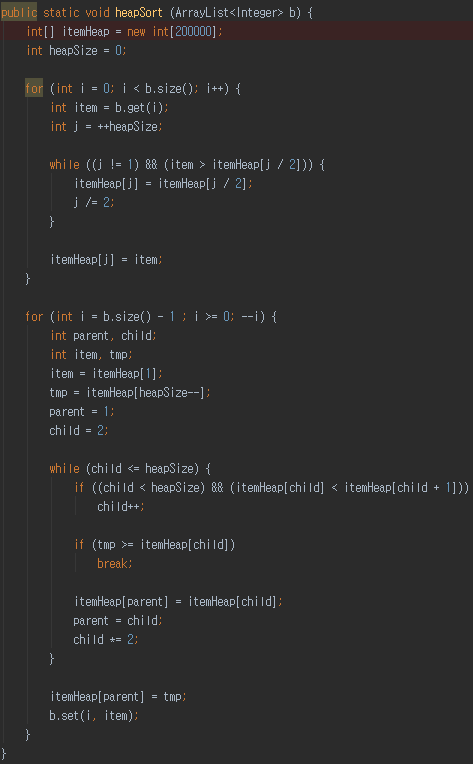
Selection sort is the easiest sorting algorithm. But it is also the slowest, because it always takes time. (Other sorting algorithms take ) time)

2. Quick sort



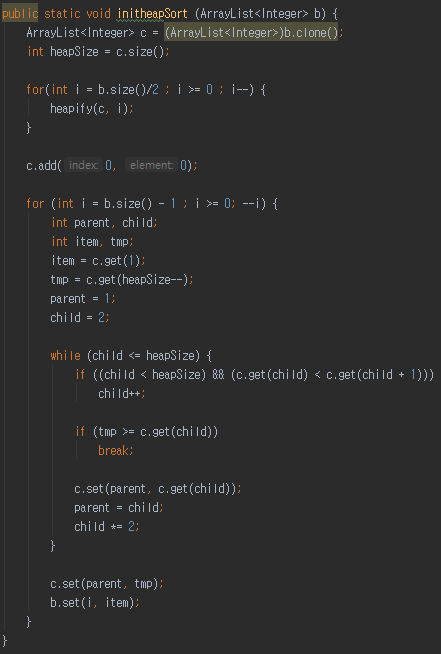
In average case and best case, it takes ) time. But in worst case, it takes ) time. Usually, quick sort is fastest because it is cache friendly than other algorithms.

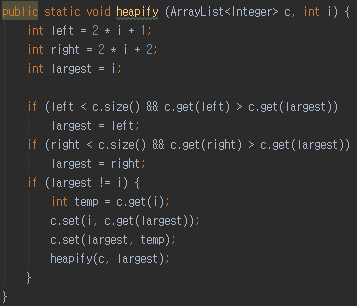
3. Heap sort (not initialized)



When using heap sort without initialization, it takes ) to offer elements and ) to poll elements.

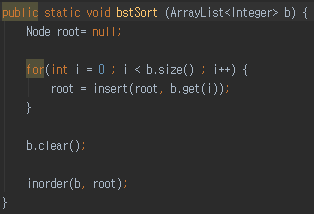
4. Heap sort (initialized)

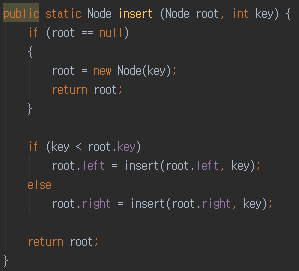


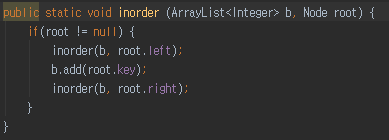


When using heap sort with initialization, it takes ) to heapify and ) to poll elements. In theory, heap sort with initialization should be faster than heap sort without initialization, but my code doesn’t (…).

5. BST sort

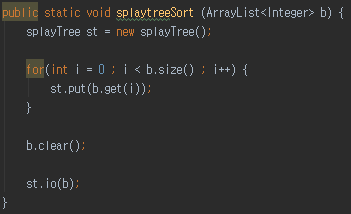






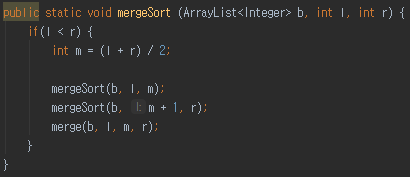
In binary search tree sort, it takes ) time in average case. But in worst case, it could take ) because tree can be unbalanced.

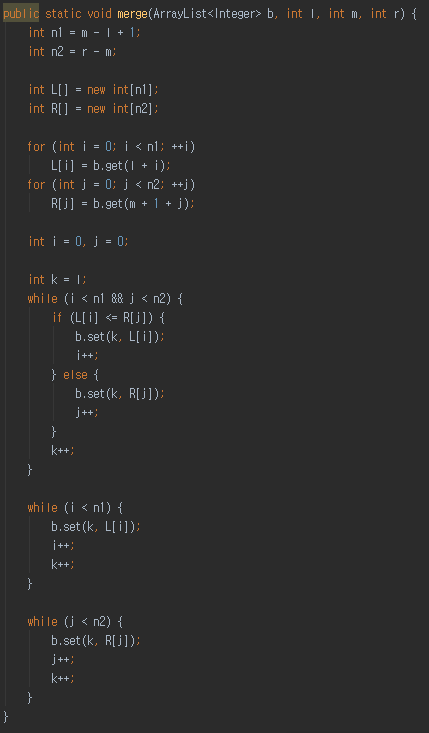
6. Splay tree sort



Splay tree sort is like BST sort, so it takes ) time to insert. Plus, it takes ) extra time to rotate tree.

7. Merge sort





Merge sort takes ) time, and it is stable sorting algorithm. But it needs extra memory to store temporary data.