

Exercise 5a. (Dynamically allocated linked list, 1p)

A reference to the first node is capable to represent the whole list in dynamic memory, provided that the next member in the last node is NULL to indicate the end of the list. In this case the variables of class LinkedList can be defined as follows:

```
public class LinkedList<T extends Comparable<T>> {  
    private Node first;  
    ...  
}
```

When operation functions are defined for that type we get the class LinkedList. Write the following operation functions for the class LinkedList defined above

```
public LinkedList();  
public void add(T item);  
public String toString();
```

Start working from the given example program (LinkedList.java, available from the Tube-portal). This list is a genuine ADT including the test application. First check how the program works, then modify the class definition by removing the `last` variable in the LinkedList class definition and modify the list implementation in such a way that it works again.

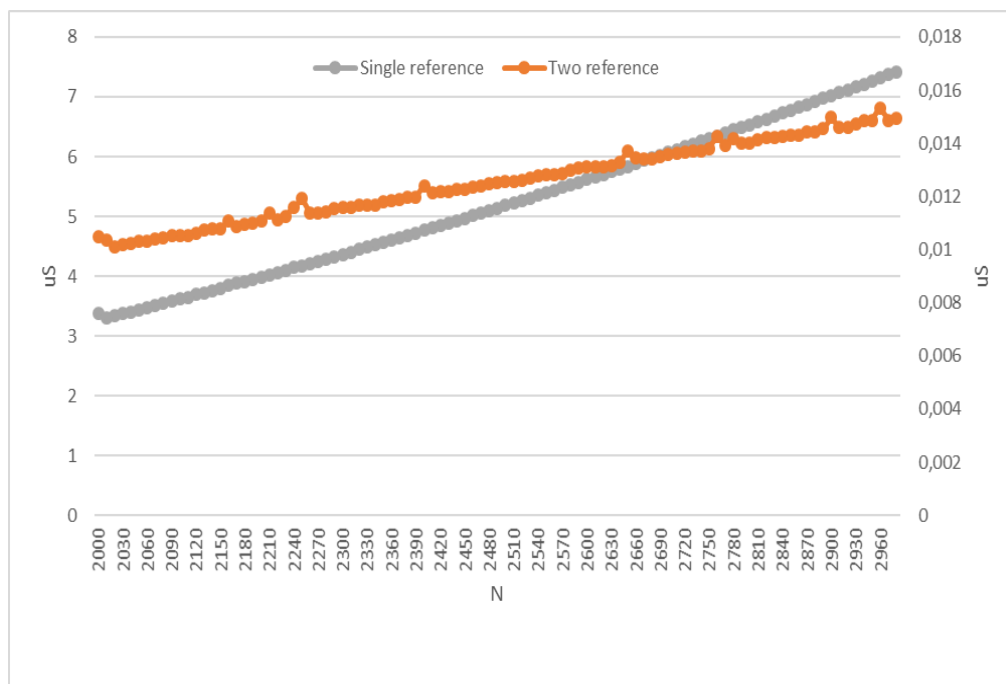
Exercise 5b. (Extra exercise, Performance measurement of two different linked list implementations, 0,5p)

This is not an easy exercise.

Insert 1000 – 2000 random integer values to the original linked list implementation (links to both the head and tail of the linked list) and to the linked list implementation of exercise 5a (link only to the head of the list). Measure the execution time of the `add()` method of both implementations using `System.nanoTime()` Java library method. There is `FunctionEvaluator.java` class given which helps statistical analysis of the execution time measurements. Put your results to the .csv¹ format for e.g. MS-Excel and plot your results against different amount of integers inserted to the list.

You should get something like:

¹ csv (comma separated values) is one of the most widely used data exchange format between different applications. In this format, data values are separated with comma (','). In Europe, comma is usually used for separating integer and fractional values (e.g. 3,1415), therefore semicolon (';') is used here instead of comma. See https://en.wikipedia.org/wiki/Comma-separated_values for further info.



This execution time measurement was done using Netbeans 8.2 on the Windows 10 and Intel i5-6600K environment with 16GB of memory². Axis on the left is for the exercise 5a implementation (single reference) and axis on the right is for the original linked list implementation (two references, i.e. link to the tail and link to the head of the list). On the average, original implementation is about 300-500 times faster.

² Notice that you get different type of graphs on different kind of machines, e.g. operating system, processor type, amount of memory has their effect to the execution speed.