Algorithms and Data Structures (ADS2)

Assessed Exercise 2

This exercise is for submission using Moodle and counts for 10% of the total assessment mark for this course.

This exercise is worth a total of 30 points.

The deadline for submission is Friday 24 March 2023 at 4:30pm.

Exercise

The exercise consists of three parts covering Queues, Stacks, and Hashing. Note you are not allowed to rely on Java library classes in your implementation.

Submission

Submit a zip archive containing the Java sources of your implementations **and** a short (max 3 pages) report describing what you have done in each part of the exercise. Your report should include a heading stating your full name and matriculation number and clear instructions on how to run your code.

Make sure the report is in pdf format and your sources are not password protected.

Part 1

Implement in Java the Queue abstract data type using a singly-linked list with only one access pointer (instead of two: Q.head and Q.tail). ENQUEUE and DEQUEUE operations should take constant time. Assume elements are integers. Briefly describe your implementation in the report.

Part 2

Design and then implement in Java a stack-like data structure to PUSH elements to the stack and POP the most frequent element from the stack. If there is a tie for the most frequent element, the element closest to the stack's top is removed and returned. Assume elements are integers. Briefly describe your implementation in the report.

Part 3

Implement in Java a function that takes as input an integer $n \ge 2$ and generates n different strings all hashing to the same value when the following hash function is used.

```
public int hashCode(String s) {
  int hash = 0;
  int skip = Math.max(1, s.length() / 3);
  for (int i = 0; i < s.length(); i += skip)
    hash = (hash * 37) + s.charAt(i);
  return hash;
}</pre>
```

Assume ASCII characters. Briefly describe your implementation in the report.

[14]