2. Werkcollege 2

You are allowed to answer in Dutch.

2.1. Show how to implement a queue using two stacks. Analyze the running time of the queue operations.

2.2. Whereas a stack allows insertion and deletion of elements at only one end, and a queue allows insertion at one end and deletion at the other end, a deque (double-ended queue) allows insertion and deletion at both ends. Write four $\mathcal{O}(1)$-time procedures to insert elements into and delete elements from both ends of a deque implemented by an array.

2.3. Implement the operations Insert, Delete, and Search using singly linked, circular lists. What are the running times of your procedures?

2.4. During the lecture, the procedures for inserting, deleting and searching in an open address hash table were not discussed in detail. Deletion is the difficult operation to implement. A Delete operation cannot simply mark a slot as empty (or NIL), because this will isolate records further down the probe sequence. We also do not want to make positions in the hash table unusable because of deletion. Both problems can be resolved by placing a special mark in place of a deleted record, called a *tombstone*. The Insert procedure may treat tombstones as if they were empty, and the search procedure may pass over tombstones while searching. Write pseudocode for the Insert, Delete, and Search procedures.

2.5. Suppose that a dynamic set S is represented by a direct-address table T of length m. Describe a procedure that finds the maximum element of S. What is the worst-case performance of your procedure?

2.6. Demonstrate what happens when we insert the keys 12, 44, 13, 88, 23, 94, 11, 39, 20, 16, 5 into a hash table with collisions resolved by chaining. Let the table have 7 slots, and let the hash function be $h(k) = k \mod 7$.