C++ Exercise 4: Running time and abstract data types

Deadline: Friday, October 9, 2020 by 23:59 EEST Timezone.
How to submit: GitHub
Grading: This exercise is worth 2% of the total grade.

1. **Running time.** Find the best-case and worst-case running times for the algorithms below, by analyzing home many times each line is evaluated, in terms of the input size. The input size is n, which is the length of the input list A.

   (a) BUBBLE-SORT

```
1  for m = A.length downto 2
2      for i = 1 to m − 1
3          if A[i] > A[i + 1]
4              swap(A[i], A[i + 1])
```

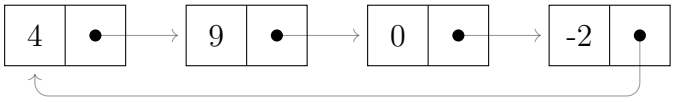
   (b) BETTER-BUBBLE-SORT

```
1  for m = A.length downto 2
2      sorted = true
3      for i = 1 to m − 1
4          if A[i] > A[i + 1]
5              swap(A[i], A[i + 1])
6              sorted = false
7      if sorted = true
8          return
```

2. **Abstract data types.**

   (a) Implement the *circular linked list*, an abstract data type where:

   - each node contains its value and a link to the next node
   - the link on the last node points back to the first node.



   (b) Implement the *circular linked double list*, an abstract data type where:

   - each node contains its value and links to the nodes below and to the right
   - the link on the last node in each row / column points back to the first node in the row / column

   You may use the `vector` STL to implement this abstract data type.