

## EECS2040 Data Structure Hw #2 (Chapter 3 Stack/Queue)

due date 4/17/2021

**Format:** Use a text editor to type your answers to the homework problem. You need to submit your HW in an HTML file or a DOCX, pdf file named as **Hw2-SNo.docx**, **Hw2-SNo.pdf** or **Hw2-SNo.html**, where SNo is your student number. Send the **Hw2-SNo.doc** or **Hw2-SNo.html** file in eLearn. Inside the file, you need to put the **header and your student number, name (e.g., EE2410 Data Structure Hw #2 (Chapter 3 of textbook) due date 4/17/2021 by SNo, name)** first, and then the **problem** itself followed by your **answer** to that problem, one by one. The grading will be based on the correctness of your answers to the problems, and the **format**. Fail to comply with the aforementioned format (file name, header, problem, answer, problem, answer,...), will certainly degrade your score. If you have any questions, please feel free to ask me.

### Part 1 (50%)

1. (30%) A linear list is being maintained circularly in an array with front and rear set up as for circular queues.
  - (a) Obtain a formula in terms of the array capacity, front, and rear, for the number of elements in the list.
  - (b) Assume the kth element in the list is to be deleted, the elements after it should be moved up one position. Give a formula describing the positions of those elements to be moved up one position, i.e., from ??? to ???.
  - (c) Assume that we want to insert an element y immediately after the kth element and array doubling is needed (as **Program 3.11** shows or pptx page 83 **void Queue<T>::Push(const T& x)** shows), please explain the code using a graphical illustration and explanation.
2. (20%) Using the operator priorities of Figure 3.15 (or pptx page 130 **Parentheses Handling**) together with those for '(' and '#' to answer the following:
  - (a) In function Postfix (Program 3.19, pptx **Infix to Postfix Algorithm**), what is the maximum number of elements that can be on the stack at any time if the input expression has n operators and delimiters?
  - (b) What is the answer to (a) if the input expression e has n operators and the depth of nesting of parentheses is at most 6?
3. (50%) Write the postfix form and prefix form of the following infix expressions:
  - (a)  $-A + B - C + D$
  - (b)  $A * -B + C$

- (c)  $(A + B) * D + E / (F + A * D) + C$
- (d)  $A \&\& B \parallel C \parallel !(E > F)$
- (e)  $!(A \&\& !((B < C) \parallel (C > D))) \parallel (C < E)$

## Part 2 Coding (50%)

You should submit:

- (a) All your source codes (C++ file).
- (b) Show the execution trace of your program.

1. (30%) Based on the circular queue and template queue ADT in **ADT 3.2** in textbook (or pptx pp.79), write a C++ program to implement the queue ADT. Then add two more functions to
  - (a) Return the size and capacity of a queue.
  - (b) Merge two queues into a one by alternately taking elements from each queue. The relative order of queue elements is unchanged. What is the complexity of your function?

You should **demonstrate all the functions** using at least one example.

2. (35%) Referring to **Program 3.13** in textbook (pptx pp.94),
  - (a) Implement Stack as a publicly derived class of Bag using template. **Demonstrate** your C++ code using at least two element types (e.g., int, float,...). **Show results** of a series of Pushes and Pops and Size functions.
  - (b) Implement Queue as a publicly derived class of Bag using template. **Demonstrate** your C++ code using at least two element types (e.g., int, float,...). **Show results** of a series of Pushes and Pops and Size functions.
  - (c) A template double-ended queue (deque) is a linear list in which additions and deletions may be made at either end. Implement the class Deque as a publicly derived templated class of Queue. The class Deque must have public functions (either via inheritance from Queue or by direct implementation in Deque) to add and delete elements from either end of the deque and also to return an element from either end. The complexity of each function (excluding array doubling) should be  $\Theta(1)$ . **Demonstrate** your C++ code using at least two element types (e.g., int, float,...). **Show results** of a series of two types of Pushes and Pops and Size functions to illustrate your code is working.

3. (35%) Write a C++ program to implement the maze in textbook using the example



---

**Figure 3.14:** 唯一路徑的迷宮圖