

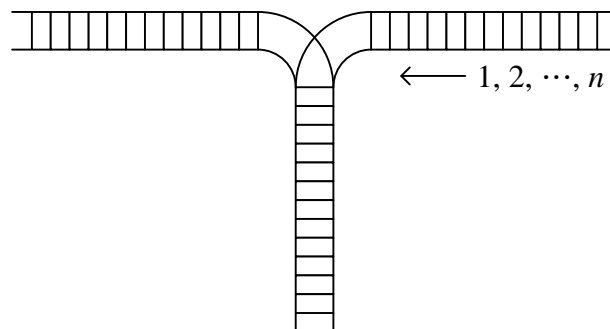
EECS2040 Data Structure Hw #2 (Chapter 3 Stack/Queue)

due date 4/11/2022

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. Submit the **Hw2-SNo.doc** or **Hw2-SNo.html** file **via eLearn**. Inside the file, you need to put the **header and your student number, name** (e.g., **EECS2040 Data Structure Hw #2 (Chapter 3 of textbook) due date 4/11/2022 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.

Part 1 (2% of final Grade)

1. (10%) Consider the railroad switching network shown below. (textbook pp.138-139)



Railroad cars can be moved into the vertical track segment one at a time from either of the horizontal segments and then moved from the vertical segment to any one of the horizontal segments. The vertical segment operates as a **stack** as new cars enter at the top and cars depart the vertical segment from the top. Railroad cars numbered 1,2,3...,n are initially in the top right track segment. Answer the following questions for **n=3** and **4** cases:

- (a) What are the possible permutations of the cars that can be obtained?
 - (b) Are any permutations not possible? If no, simply answer no. If yes, list them all.
2. (15%) A linear **list** of type T objects is being maintained circularly in an array with front and rear set up as for **circular queues**.
 - (a) Draw a diagram of the circular array showing the initial status (values of front, rear) when the linear list is created (constructed) with no elements stored yet

- (assume capacity = 10 is used for creating the array)
- (b) Obtain a formula in terms of the array capacity, front, and rear, for the number of elements in the list.
 - (c) 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 in terms of k, front, rear, capacity.
Design an algorithm (pseudo code) for this Delete(int k) member function.
 - (d) Assume that we want to insert an element y immediately after the kth element. So the elements from the (k+1)th element on should be moved down one position in order to give space for y, which might cause insufficient capacity case in which array doubling will be needed. Describe the situation when array doubling is needed (in terms of k, front, rear, capacity). Design an algorithm (pseudo code) for this Insert(int k, T& y) member function.
 - (e) The following code segment is used for Push member function in circular queue described in textbook. Please explain the code in detail. (using a graphical illustration and explanation)

```

template <class T>
void Queue<T>::Push(const T& x)
{
    // add x to queue
    if ((rear + 1) % capacity == front) //resize
    {
        T* newQu = new T[2*capacity];
        int start = (front+1) % capacity;
        if(start<2)
            copy(queue+start, queue+start+capacity-1, newQu);
        else{
            copy(queue+start, queue+capacity, newQu);
            copy(queue, queue+rear+1,newQu+capacity-start);
        }
        front = 2*capacity - 1;
        rear = capacity -2;
        delete[] queue;
        queue = newQu; capacity *=2;
    }
    rear = (rear+1)%capacity;  queue[rear] = x;
}

```

3. (10%) Design an algorithm, reverseQueue, that takes as a parameter a queue

object and uses a stack object to reverse the elements of the queue. The operations on queue and stack should strictly follow the ADT 3.2 Queue ADT and ADT 3.1 Stack ADT.

4. (5%) Given an integer k and a queue of integers, how do you reverse the order of the first k elements of the queue, leaving the other elements in the same relative order? For example, if $k=4$ and queue has the elements [10, 20, 30, 40, 50, 60, 70, 80, 90]; the output should be [40, 30, 20, 10, 50, 60, 70, 80, 90]. Design your algorithm for this task.
5. (10%) Suppose that you are a financier and purchase 100 shares of stock in Company X in each of January, April, and September and sell 100 shares in each of June and November. The prices per share in these months were

| | | | | |
|------|------|------|------|------|
| Jan | Apr | Jun | Sep | Nov |
| \$10 | \$30 | \$20 | \$50 | \$30 |

Determine the total amount of your capital gain or loss using (a) FIFO (first-in first-out) accounting and (b) LIFO (last-in, first-out) accounting [that is, assuming that you keep your stock certificates in (a) a queue or (b) a stack.]

The 100 shares you still own at the end of the year do not enter the calculation.

6. (15%) For the maze problem,
 - (a) What is the maximum path length from start to finish for any maze of dimensions $m \times p$?
 - (b) Design a recursive version of algorithm for Path().
 - (c) What is the time complexity of your recursive version?
7. (10%) Using the operator priorities of Figure 3.15 (shown below) together with those for '(' and '#' to answer the following:

| priority | operator |
|----------|----------------|
| 1 | Unary minus, ! |
| 2 | *, /, % |
| 3 | +, - |
| 4 | <, <=, >, >= |
| 5 | ==, != |
| 6 | && |
| 7 | |

- (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?
8. (20%) Write the postfix form and prefix form of the following infix expressions:
- (a) $-A + B - C + D * A / B$
- (b) $A * -B + C / D - A * C$
- (c) $(A + B) / D + E / (F + A * D) + C$
- (d) $A \&\& B \parallel C \parallel !(E > F)$
- (e) $!(A \&\& !((B < C) \parallel (C > E))) \parallel (C < D)$
9. (10%) Evaluate the following postfix expressions:
- (a) $8\ 2\ +\ 3\ *\ 16\ 4\ /\ -\ =$
- (b) $12\ 25\ 5\ 1\ /\ *\ 8\ 7\ +\ -\ =$