Homework 2

Due date: Jan 23, 2020, 9:30am

Objective

- Trace operations on stacks, queues, and deques.
- Implement algorithms using stacks and queues.
- Determine which data structure / abstract data type is most suitable to solve a given problem.
- Trace depth-first search.

Exercises

1. (10 points)
   Trace the following program segment given in pseudocode. List the elements in `queue` at the end of the program segment from head to tail. List the elements in `stack` at the end of the program segment from bottom to top.

```
List list = new empty linked list of integers;
Stack stack = new empty stack of integers;
Queue queue = new empty queue of integers;

for (int counter = 1; counter <= 3; counter++) {
   list.addTail(counter);
}
```
// content of list from head to tail: _____

```
for (int counter = 4; counter <= 6; counter++) {
   list.addHead(counter);
}
```
// content of list from head to tail: _____

```
while (list is not empty) {
   stack.push(list.getTail());
   list.removeTail();
}
```
// content of stack from bottom to top: _____

// content of list from head to tail: _____

```
Integer number = stack.pop();
queue.enqueue(stack.pop());
```

```
queue.enqueue(number);
queue.enqueue(stack.pop());
queue.enqueue(queue.dequeue);
queue.enqueue(queue.dequeue);
```
// content of stack from bottom to top: _____

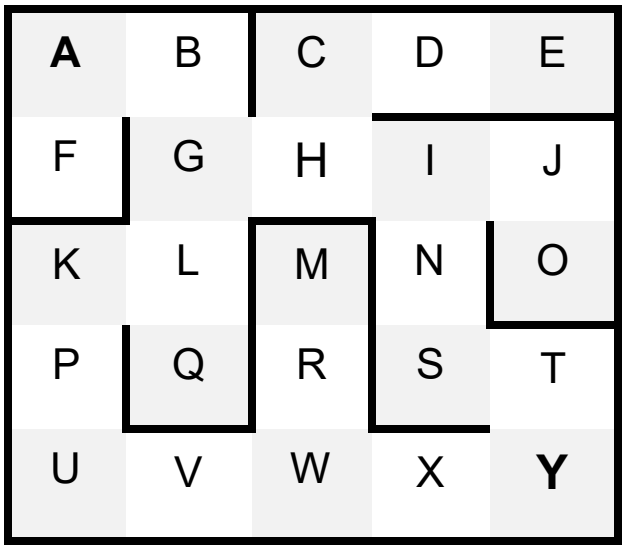
// content of queue from head to tail: _____

2. (10 points)
   Determine which data structure is most suitable in different scenarios:
   a) A web browser remembers the web pages you have visited. This information is used when you go back one page. Note that you can go back one page repeatedly. Which abstract data type is most suitable to store the visited web pages for the operation that allows the user to go back one page, possibly repeatedly?
   b) Consider the basic version of the domino game where domino tiles are played in a series of tiles. Adjacent tiles must have matching values. A player can only add tiles at one of the two ends of the line of tiles. What abstract data type would be most suitable to store the played tiles?

3. (10 points)
   Write pseudocode that reverses the order of the elements in a queue using one additional queue or one stack, and possibly some variables. Do not use any other additional collection. The only valid stack operations are isEmpty, push, pop, and peek. The only queue operations are isEmpty, enqueue, dequeue, and peek.

4. (10 points)
   Write pseudocode that reverses the order of the elements in a stack using one additional stack and possibly some variables. Do not use any other additional collection. The only valid stack operations are isEmpty, push, pop, and peek. You are not allowed to use a pointer or reference for a stack, that is, an operation of the form “stack1 = stack2;” is not valid where stack1 and stack2 are stacks.

5. (10 points)
   Below is a maze where each cell is labeled by a letter. Each cell can be modeled as a node. The goal is to determine a path from the source node A to the destination node Y. Apply DFS by using the generic search algorithm where the collection `nextNodes` is implemented as a stack. The neighbors of a cell are traversed clockwise starting at the upper cell. For example, if the neighbors of cell H are added to the stack in case of DFS, they are added in the order C, I, G. In particular, G will be at the top of the stack and therefore the search will continue removing G from nextNodes in the next iteration.



a) List the order of the nodes in which they are removed from the collection nextNodes in case of DFS.

b) What is the solution path determined by DFS?

Submission

Turn in a hard-copy with your solutions at the beginning of the class meeting on Jan 23. The solution can be hand-written, or typed & printed. A printer is available in the Advanced Computing Lab.