Linear Data Structures

Anna Xu

10/6/17

1 Introduction

Linear data structures are used to store and organize data and when implemented effectively, can reduce the space and time complexities of different tasks. The four most common examples are arrays/arraylists, linked lists, stacks, and queues.

2 Arrays/ArrayLists

2.1 Arrays

```
int [] a = new int[20];
int [][] grid = new int[20][20];
int [][][][][] space = new int [5][5][5][5][5];
```

2.2 ArrayLists

ArrayList in Java is a data structure that can be modified to accommodate the number of items stored. It differs from an array in that the size does not need to be determined upon initialization; the add() and remove() methods can be used to change the size. Instead of array.length, the size is arraylist.size().

3 Linked Lists

Linked lists are a dynamic data structure where each data item in the lists points to the address of the next data item in the list. Advantages include easier insertions and deletions and efficient memory utilization but searching for a particular element is difficult and time consuming.

3.1 Types of Linked Lists

1. Single linked list (may be circular)
2. Double linked list (may be circular)

A singular linked list is one in which all nodes are linked together in a sequential manner. A double linked list is one in which all nodes are linked together by multiple links which helps in accessing both the successor node (next node) and predecessor node (previous node) from any arbitrary node within the list. Therefore each node in a double linked list has two link fields (pointers) to point to the left node (previous) and the right node (next). This helps to traverse in forward direction and backward direction.

4 Stacks/Queues

Stacks and queues are data structures that allow us to insert and remove items, but they differ in deletion order.

4.1 Stacks

Stacks are a LIFO data structure– the last item added to the stack is the first item out of the stack. All stack operations, including creation, checking if it is empty, pushing items, and popping items run in O(1) time. Stacks are useful for backtracking problems like going through a maze where you may need to reverse the most recent moves and keep going as well as in Depth First Search (DFS).

4.2 Queues and PriorityQueues

Queues are a FIFO structure– the first item added is the first item out. Queues are used in implementing Breadth First Search (BFS). The priority queue is a special type of queue where the items are arranged in decreasing order of priority. The most important item will be at the front regardless of insertion time and will be popped first. Example: a lunch line of students where the seniors are in front, followed by juniors, sophomores, and freshmen. PriorityQueue placement is determined by the compareTo() method in Java.

4.3 Example

What is the state of the stack or queue after each operation?

```
push(5);
push(9);
push(3);
push(pop() + pop());
push(4);
push(2);
push(pop() * pop());
push(pop() + pop());
pop();
isEmpty()
```

5 Problems

Problem 1: Check for balanced parenthesis, braces, and other paired delimiters in text.

Problem 2: Use stacks to solve the "N queens" problem. (Place N queens on an N x N chess board so that no two queens are on the same row, column, or diagonal.)

Problem 3 (USACO Jan 2017 Contest, Silver): Farmer John's N cows are standing in a row, as they have a tendency to do from time to time. Each cow is labeled with a distinct integer ID number so FJ can tell them apart. FJ would like to take a photo of a contiguous group of cows but, due to a traumatic childhood incident involving the numbers $1 \ldots 6$, he only wants to take a picture of a group of cows if their IDs add up to a multiple of 7. Please help FJ determine the size of the largest group he can photograph.