Course: CSC220.01

Student: Mark Kim

Instructor: Duc Ta

Assignment Number: 03

Due Date & Time: 07-31-2019 at 11:55 PM

Assignment 03

PART A – Introduction to Sorting

A. Selection Sort

9	5	7	8	3	2	4	7	6	1
1	5	7	8	3	2	4	7	6	9
✓	2	7	8	3	5	4	7	6	9
	✓	3	8	7	5	4	7	6	9
		✓	4	7	5	8	7	6	9
			✓	5	7	8	7	6	9
				✓	6	8	7	7	9
					✓	7	8	7	9
						✓	7	8	9
							✓	8	9
								✓	✓
1	2	3	4	5	6	7	7	8	9

B. Insertion Sort

9	5	7	8	3	2	4	7	6	1
9	5								
5	9	7							
5	7	9	8						
5	7	8	9	3					
3	5	7	8	9	2				
2	3	5	7	8	9	4			
2	3	4	5	7	8	9	7		
2	3	4	5	7	7	8	9	6	
2	3	4	5	6	7	7	8	9	1
1	2	5	4	5	6	7	7	8	9

C. Shell Sort

9	5	7	8	3	2	4	7	6	1	10/2=5
2					9					
	4					5				
		7					7			
			6					8		
				1					3	
2	4	7	6	1	9	5	7	8	3	5/2=2
1		2		5		7		8		
	3		4		6		7		9	
1	3	2	4	5	6	7	7	8	9	2/2=1
1	3									
1	3	2								
1	2	3	4							
1	2	3	4	5						
1	2	3	4	5	6					
1	2	3	4	5	6	7				
1	2	3	4	5	6	7	7			
1	2	3	4	5	6	7	7	8		
1	2	3	4	5	6	7	7	8	9	

D. Compare time efficiency of two algorithms

Algorithm A searches the entire array sequentially and records the largest entry seen so far, while algorithm B sorts the array in descending order and then reports the first entry as the largest. Algorithm A will be much faster than algorithm B because A only needs to traverse the data set at most once to find the largest entry, so it will do at most n comparisons to find the largest value. For algorithm B, it must traverse the data multiple times to find the data order and also perform operations to move the data into descending order.

E. Algorithm to sort n by n array of integer values by first value.

This portion of the assignment tasked us to implement an algorithm to sort a 2-dimensional array into ascending order of the first value. Using the provided skeleton code, I came to the conclusion that the simplest solution would be to implement a selection sort to

complete the task. In addition, an empty swap method was provided, which is only used by selection sort from the sort methods taught to us at this juncture. The task simply required us to traverse the data by the initial index and swap the entire row into the final position. This was done by copying the contents of the row using a for loop and a temporary variable to swap the data in the row for the data that was to be put in its final position.

Initially, I misunderstood the assignment and thought that each row was supposed to also be sorted in ascending order starting with the number in the initial position. This would've been a much more difficult task and I abandoned this task once I figured out that this was not what the assignment was asking.

PART 2 – Queues, Deques, and Priority Queues

A. Queue

1,2,3	Jane	Jess	Jon		
4	Jess	Jon	Jane		
5	Jess	Jon	Jane	Jess	
6	Jess	Jon	Jane	Jess	Jim
7	Jon	Jane	Jess	Jim	
8	Jon	Jane	Jess	Jim	Jon

B. Deque

1	Jim					
2	Jess	Jim				
3	Jess	Jim	Jen			
4	Jess	Jim	Jen	Josh		
5	Jim	Jen	Josh			
6	Jim	Jen	Josh	Jess		
7	Jim	Jen	Josh	Jess	Jim	
8	Jim	Jim	Jen	Josh	Jess	
9	Jess	Jim	Jim	Jen	Josh	Jess

C. Priority Queue

1	Jim					
2	Jim	Josh				
3	Jim	Jon	Josh			
4	Jane	Jim	Jon	Josh		
5	Jim	Jon	Josh			
6	Jane	Jim	Jon	Josh		
7	Jane	Jane	Jim	Jon	Josh	
8	Jane	Jane	Jim	Jon	Jose	Josh
9	Jane	Jim	Jon	Jose	Josh	

D. Implement Circular Doubly Linked Chain ADT deque

In this portion of the assignment, we were tasked to implement a circular doubly linked chain deque. The purpose of this portion of the assignment was to get a better understanding of circular chains, deques, and doubly linked chains. I have to admit that I was stumped at first. One student was able to describe his implementation of this assignment, but it really did not help me understand how to start this portion of the assignment.

What allowed me to understand and start the implementation of this deque was to diagram the chain from initiation to the addition of the second and third nodes. Once I was able to grasp the movement of the pointers, I was able to track what portion of each node needed to point to after the addition of a new node.

Zip file contents:

MarkKim-Assignment-03-Report.pdf	(this report)
asmt03Part1E	(folder containing all java files for part 1E)
asmt03Part2D	(folder containing all java files for part 2D)