

## ECE 71/191T – Data Structures and Algorithms

Dr. Gregory R. Kriehn, Fresno State  
C++ Homework Assignment: Chapter 20

**Code Due By:** Midnight on Wed, May 03

**Write-up Due By:** Class on Thu, May 04

### HOMEWORK #36 – Depth and Breadth Tree Traversal

Implement the `graphType.h` template class, as outlined in Chapter 20 of Malik. You will also need to incorporate the `linkedList.h`, `unorderedLinkedList.h` (with iterators), `queueADT.h`, and `linkedQueue.h` template classes, as developed from previous homework, as well. Write a program that outputs the nodes of a graph in a depth first, followed by a breadth first traversal. Create a `data1.txt` file with the following data for your program:

```
10
0 1 3 -999
1 4 -999
2 5 -999
3 2 -999
4 -999
5 7 8 -999
6 4 7 -999
7 -999
8 -999
9 7 8 -999
```

The results from your program should be:

```
Enter input filename: data1.txt
```

```
Depth First Traversal: 0 1 4 3 2 5 7 8 6 9
```

```
Breadth First Traversal: 0 1 3 4 2 5 7 8 6 9
```

### HOMEWORK #37 – Shortest Path Algorithm

Implement the `weightedGraph.h` derived template class to implement the shortest path algorithm, as outlined in Chapter 20 of Malik. Write a program that outputs the shortest distance from a given node to every other node in the graph. Create a `data2.txt` file with the following data for your program:

```
// Adjacency List
5
0 1 3 4 -999
1 2 -999
```

```

2 1 -999
3 1 4 -999
4 1 2 3 -999

// Weight List
// First column corresponds to the current vertex
// Next entry is the destination vertex
// Next entry is the weight
// Next entry is the next destination vertex
// Next entry is the next weight
// Repeat until -999
// Row 0, Col 0: Source Vertex = 0
// Row 0, Col 1: Destination Vertex = 0
// Row 0, Col 2: Weight = 0
// Row 0, Col 3: Destination Vertex = 1
// Row 0, Col 4: Weight = 16
// Etc.
// This data file implements Figure 20-7 in Chapter 20.

0 0 0 1 16 3 2 4 3 -999
1 1 0 2 5 -999
2 1 3 2 0 -999
3 1 12 3 0 4 7 -999
4 1 10 2 4 3 5 4 0 -999

```

The results from your program should be:

```
Enter input filename: data2.txt
```

```
Source Vertex: 0
```

```
Shortest Distance from Source to each Vertex.
```

```
Vertex    Shortest Distance
```

```

0          0
1          10
2          7
3          2
4          3

```

### HOMEWORK #38 – Minimal Spanning Tree Algorithm

Implement the `minimalSpanTreeType.h` derived template class to implement the shortest path algorithm, as outlined in Chapter 20 of Malik. Write a program that outputs the minimal spanning tree for a given graph. Create a `data3.txt` file with the following data for your program:

```
0 1 2 3 -999
1 0 4 6 -999
2 0 5 6 -999
3 0 4 -999
4 1 3 5 -999
5 2 4 -999
6 2 1 -999
```

```
// Same data format as the previous problem.
```

```
0 1 6 2 5 3 2 -999
1 0 6 4 2 6 4 -999
2 0 5 5 7 6 5 -999
3 0 2 4 8 -999
4 1 2 3 8 5 10 -999
5 2 7 4 10 -999
6 1 4 2 5 -999
```

The results from your program should be:

```
Enter input filename: data3.txt
```

```
Source Vertex: 0
```

```
Shortest Distance from Source to each Vertex.
```

```
Vertex    Shortest Distance
```

0	0
1	6
2	5
3	2
4	8
5	12
6	10