CS 5800: Algorithms SEC 05 Summer Full 2024 (Seattle)
Homework 5

Payal Chavan

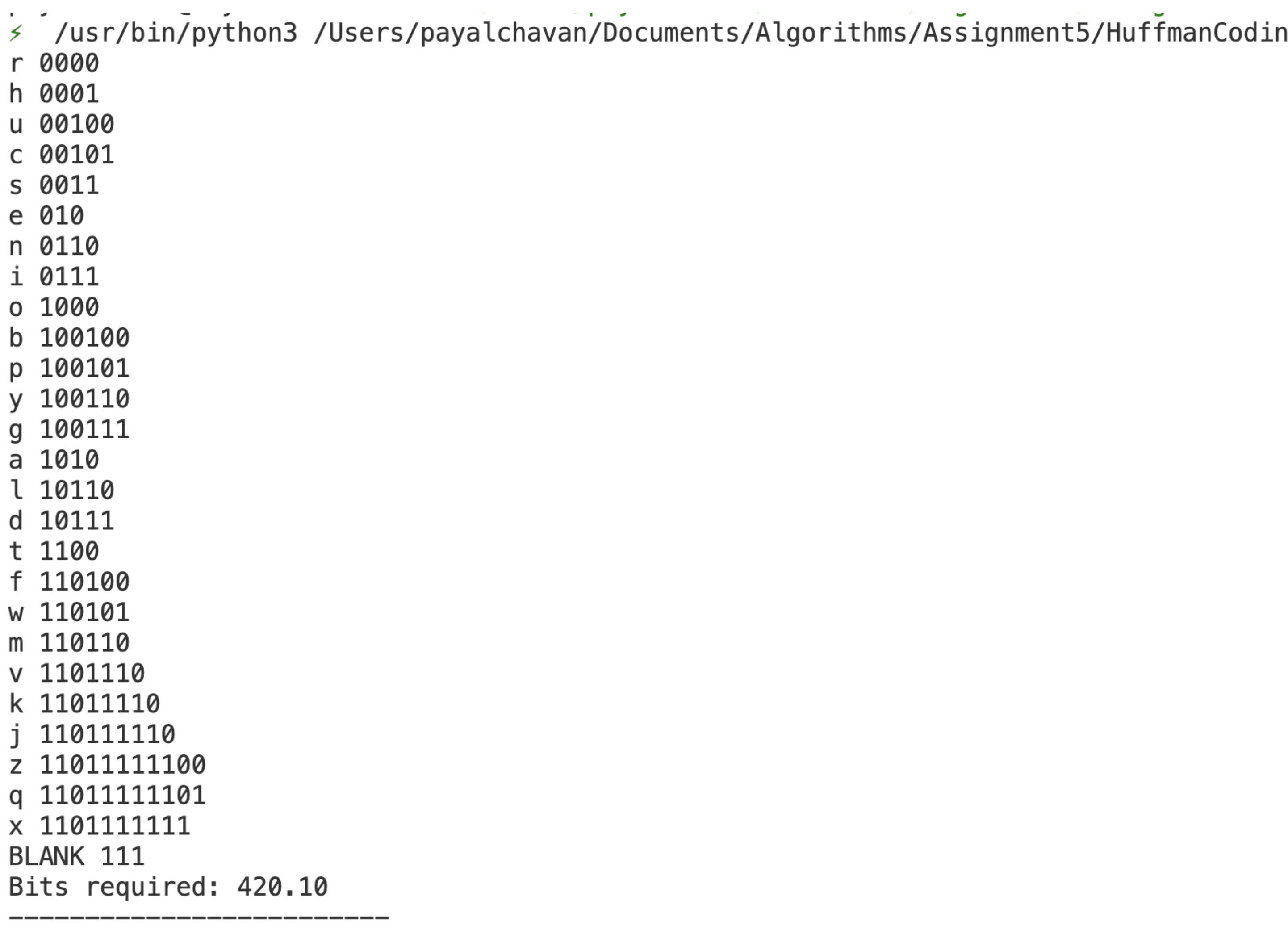
06/21/2024

Question 1: Implementation of Exercise 5.18 Huffman Encoding.

Part (a): What is the optimum Huffman encoding of this alphabet?

Part (b): What is the expected number of bits per letter?

Output:

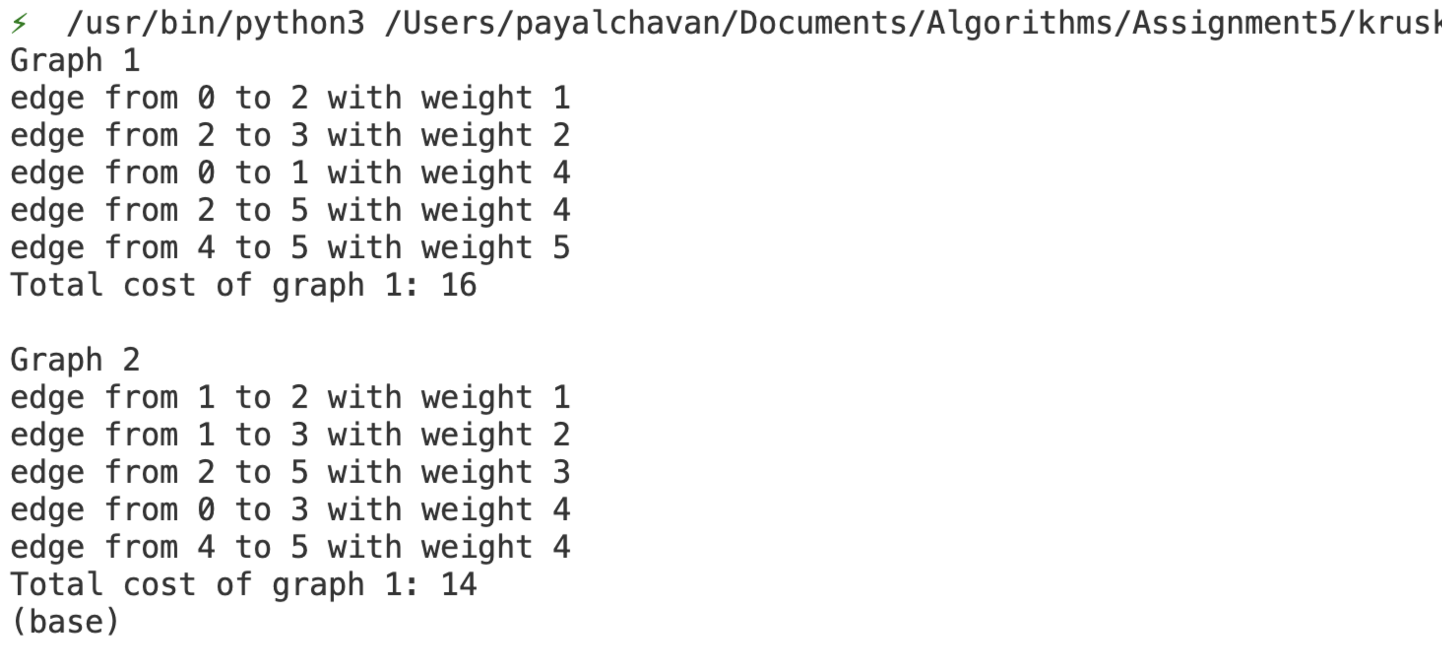


```
⚡ /usr/bin/python3 /Users/payalchavan/Documents/Algorithms/Assignment5/HuffmanCodin
r 0000
h 0001
u 00100
c 00101
s 0011
e 010
n 0110
i 0111
o 1000
b 100100
p 100101
y 100110
g 100111
a 1010
l 10110
d 10111
t 1100
f 110100
w 110101
m 110110
v 1101110
k 11011110
j 110111110
z 11011111100
q 11011111101
x 1101111111
BLANK 111
Bits required: 420.10
-------------------------
```

Figure 1: Huffman Encoding Output with no. of bits

Question 2: Implementation of Kruskal's Algorithm.

Output:



```
⚡ /usr/bin/python3 /Users/payalchavan/Documents/Algorithms/Assignment5/krusk
Graph 1
edge from 0 to 2 with weight 1
edge from 2 to 3 with weight 2
edge from 0 to 1 with weight 4
edge from 2 to 5 with weight 4
edge from 4 to 5 with weight 5
Total cost of graph 1: 16

Graph 2
edge from 1 to 2 with weight 1
edge from 1 to 3 with weight 2
edge from 2 to 5 with weight 3
edge from 0 to 3 with weight 4
edge from 4 to 5 with weight 4
Total cost of graph 1: 14
(base)
```

Figure 2: Kruskal's Algorithm Output

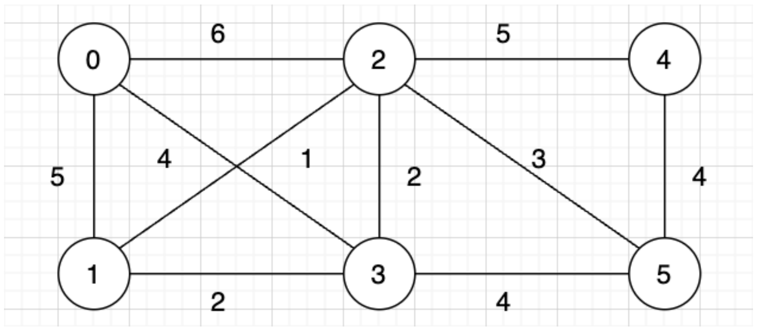


Figure 3: Original Graph

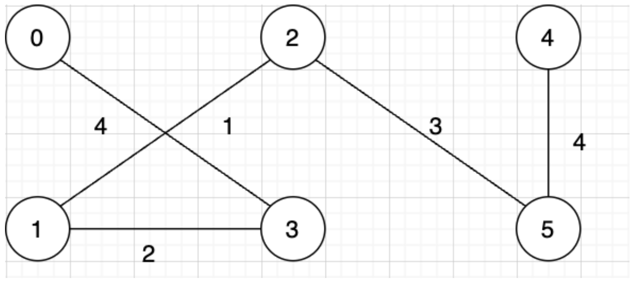
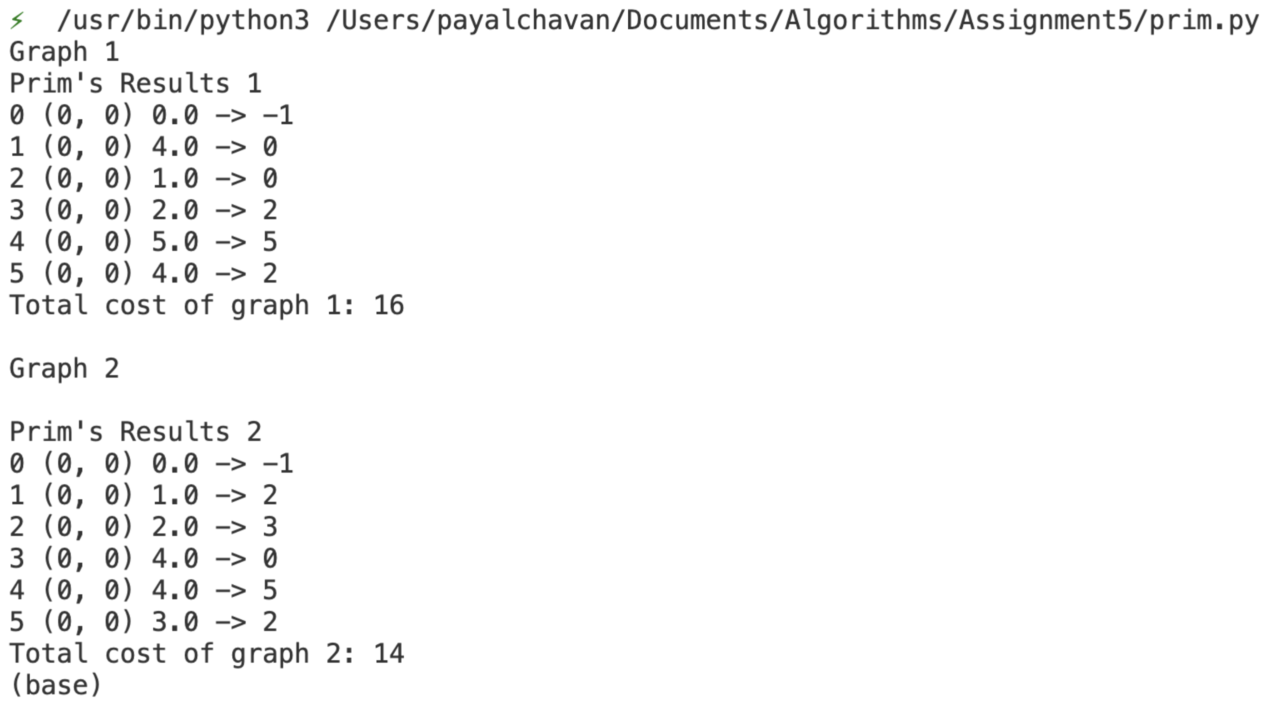


Figure 4: Kruskal Algorithm MST

From the above MST Graph, we get the total cost as $(4+2+1+3+4=14)$, which is as same from the above console output. Also, the MST does not form a cycle.

Question 3: Implementation of Prim's Algorithm.

Output:



```
⚡ /usr/bin/python3 /Users/payalchavan/Documents/Algorithms/Assignment5/prim.py
Graph 1
Prim's Results 1
0 (0, 0) 0.0 -> -1
1 (0, 0) 4.0 -> 0
2 (0, 0) 1.0 -> 0
3 (0, 0) 2.0 -> 2
4 (0, 0) 5.0 -> 5
5 (0, 0) 4.0 -> 2
Total cost of graph 1: 16

Graph 2

Prim's Results 2
0 (0, 0) 0.0 -> -1
1 (0, 0) 1.0 -> 2
2 (0, 0) 2.0 -> 3
3 (0, 0) 4.0 -> 0
4 (0, 0) 4.0 -> 5
5 (0, 0) 3.0 -> 2
Total cost of graph 2: 14
(base)
```

Figure 5: Prim's Algorithm Output

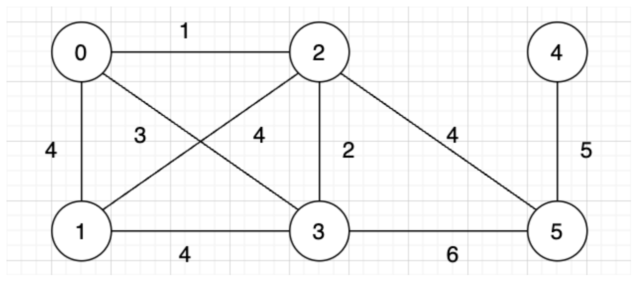


Figure 6: Original Graph1

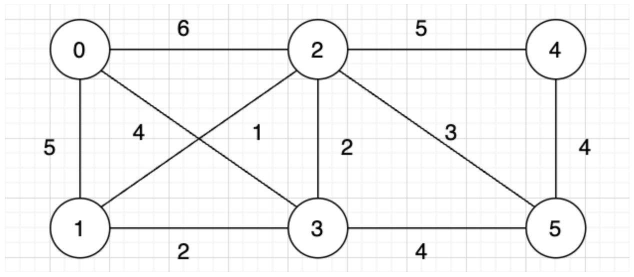


Figure 7: Original Graph2

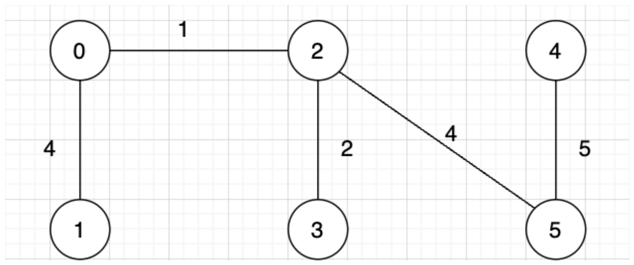


Figure 8: Graph1 - Prim's MST

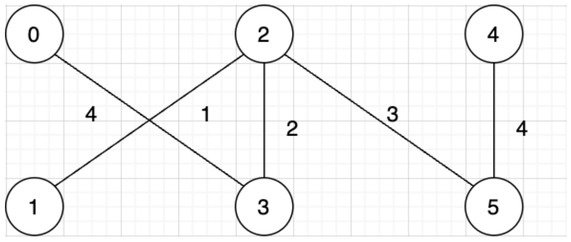


Figure 9: Graph2 - Prim's MST

For Graph1 the Prim's MST is $(4+1+2+4+5=16)$, which is similar to the above console output for Prim's Algorithm. Also, this graph does not form a cycle.

Similarly, for Graph2 the Prim's MST is $(4+1+2+3+4=14)$, which is similar to the above console output for Prim's Algorithm. Also, this graph does not form a cycle.

The Big-Oh complexity of Prim's algorithm is $\mathcal{O}((V+E)\log V)$, where V is the number of vertices and E is the number of edges.
The algorithm's performance depends on the edge density. On average, each vertex has a constant number of adjacent edges. The priority queue operations and updates can vary, affecting the average logarithmic time complexity for each operation.

Question 4: Implementation of Prim's Algorithm and Kruskal's Algorithm.

Output:

The graph with (N = 8) vertices and (E = 16) edges is shown below:

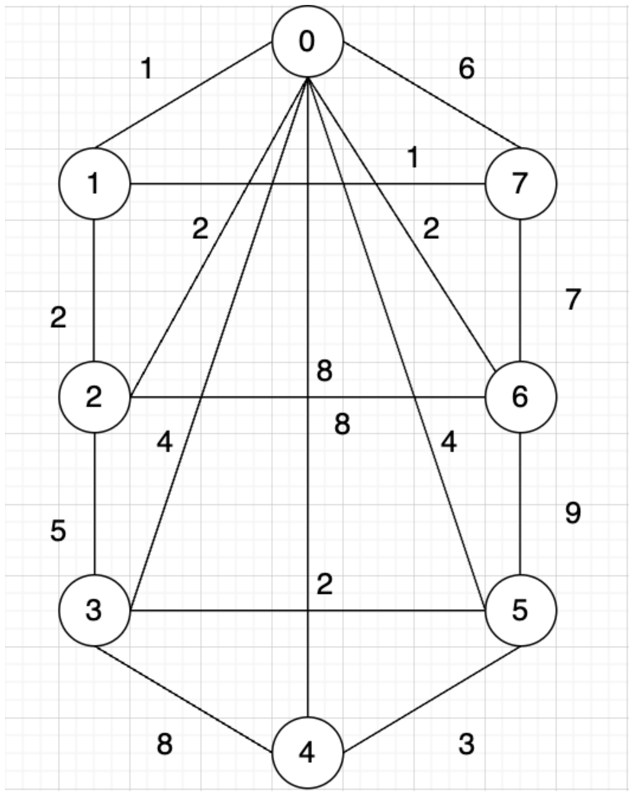


Figure 10: Original Graph

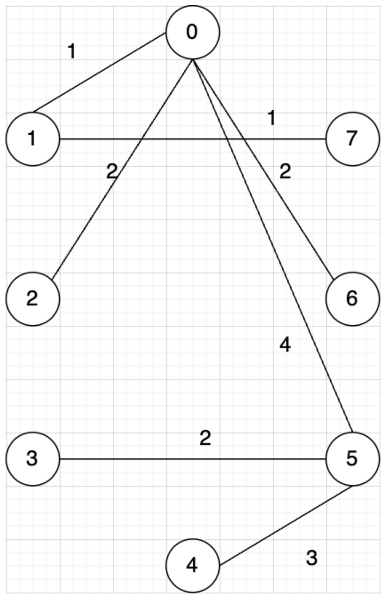


Figure 11: Prim's Minimum Spanning Tree

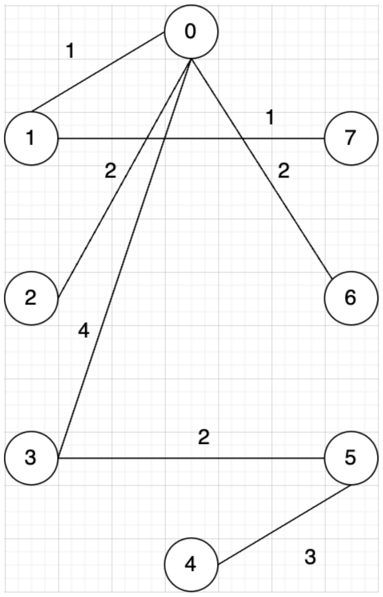
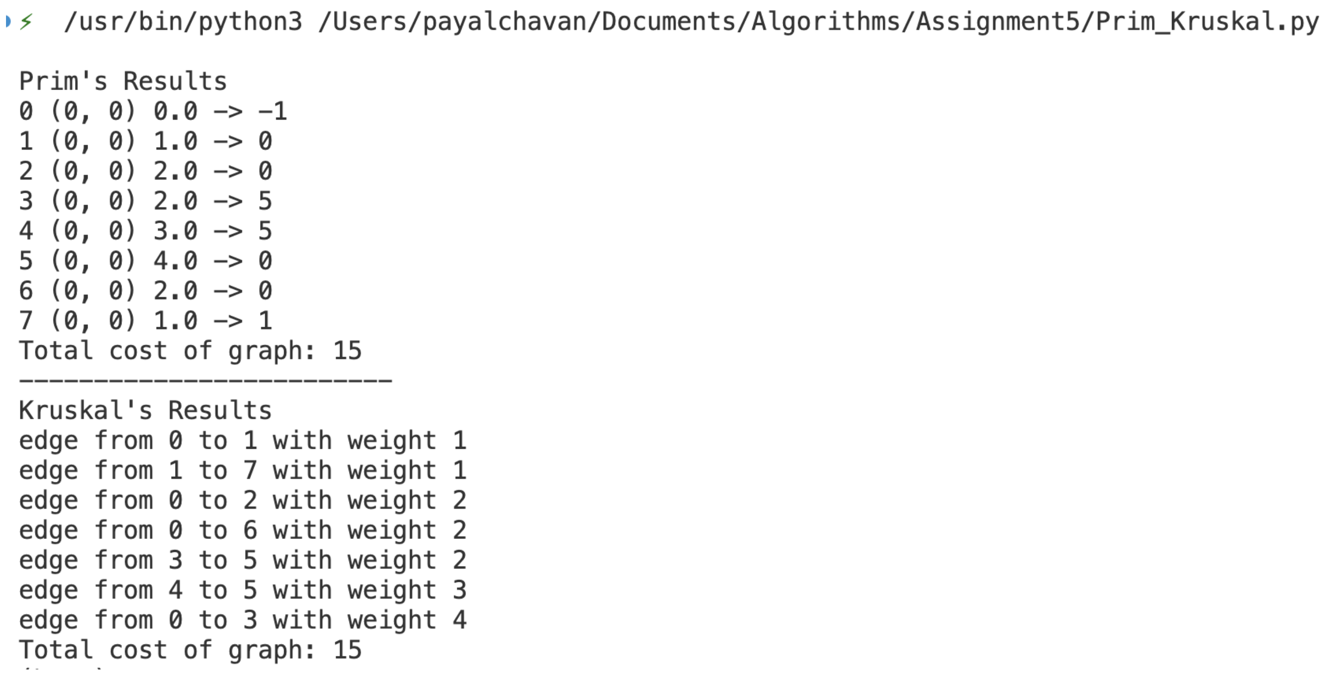


Figure 12: Kruskal's Minimum Spanning Tree



```
⚡ /usr/bin/python3 /Users/payalchavan/Documents/Algorithms/Assignment5/Prim_Kruskal.py

 Prim's Results
 0 (0, 0) 0.0 -> -1
 1 (0, 0) 1.0 -> 0
 2 (0, 0) 2.0 -> 0
 3 (0, 0) 2.0 -> 5
 4 (0, 0) 3.0 -> 5
 5 (0, 0) 4.0 -> 0
 6 (0, 0) 2.0 -> 0
 7 (0, 0) 1.0 -> 1
 Total cost of graph: 15
 -------------------------
 Kruskal's Results
 edge from 0 to 1 with weight 1
 edge from 1 to 7 with weight 1
 edge from 0 to 2 with weight 2
 edge from 0 to 6 with weight 2
 edge from 3 to 5 with weight 2
 edge from 4 to 5 with weight 3
 edge from 0 to 3 with weight 4
 Total cost of graph: 15
```

Figure 13: Output of Prim and Kruskal Algorithm

From the Kruskal's and Prim's output, we can verify that the minimum spanning trees (MST) has the number of edges equal to 7, which is $(|V| - 1 = 8 - 1 = 7)$ edges. And, it does not form a cycle as seen in the MST figure. Also, the calculation of total cost from the MST graph figure comes out to be $(1 + 2 + 4 + 1 + 2 + 2 + 3 = 15)$, which is similar to the console output.

Reflection:

From this coding exercise, I learnt how to implement Prim's Algorithm and Kruskal's Algorithm on any given undirected connected graph. These algorithms provided valuable insights into drawing minimum spanning trees (MST). Additionally, I learnt Huffman Encoding concept.

Acknowledgements:
I would like to express my sincere gratitude to the following individuals and resources for their invaluable contributions to this assignment:
1) Prof. Bruce Maxwell: Thank you, Professor Bruce Maxwell for your guidance in this assignment. Your expertise in algorithms greatly influenced my work.
2) Classmates and TA's: I appreciate the discussions of my classmates, and TA's who clarified my doubts.
3) DPV Algorithms Textbook: This textbook was a useful resource for me to understand the basics of algorithms.
4) https://www.geeksforgeeks.org/prims-minimum-spanning-tree-mst-greedy-algo-5/: This website provided clear explanations for understanding Prim's Algorithm.
5) https://www.geeksforgeeks.org/huffman-coding-greedy-algo-3/?ref=lbp: This was a useful resource to understand Huffman Encoding.
6) https://www.programiz.com/dsa/kruskal-algorithm: This website provided clear explanations for understanding Kruskal's Algorithm.