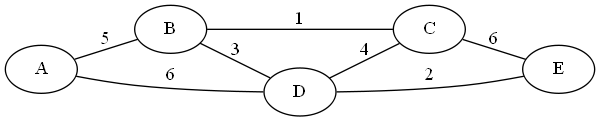
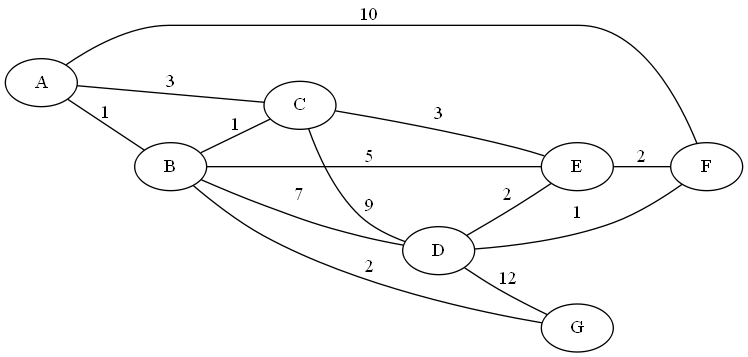
Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Apply Kruskal’s algorithm to find a minimum spanning tree of the graph below.

2. Use Dijkstra’s algorithm to find the shortest path between nodes **A** and **F**. (Start at A, end at F)

|  |  |
| --- | --- |
| **Tree Vertices** | **Remaining Vertices** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

Path: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Distance: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

3.

a. Construct a Huffman tree for the following data:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Symbol** | A | B | C | D | \_ |
| **Frequency** | 0.4 | 0.1 | 0.2 | 0.15 | 0.15 |

b. Encode ABACABAD using the tree you generated for (a).

c. Decode 100010111001010 using the tree you generated for (a).

d. What compression gain (percent of improvement) do we get by using Huffman encoding instead of a fixed‐length encoding scheme (assume the fixed‐length scheme would require 3 bits for each character)?