CS 463
Section 01
Spring 2021
Christopher K. Schmitt Assignment 6

Problem 1

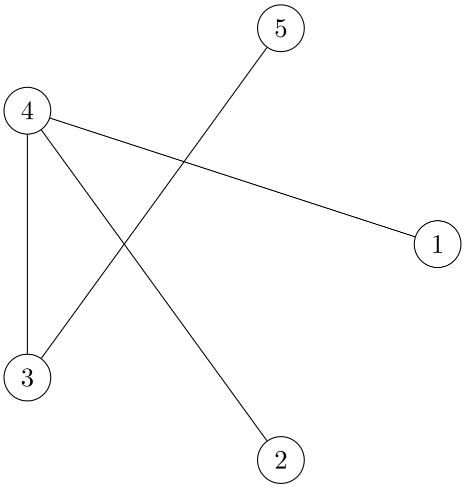
We can use a greedy approach to develop an efficient solution to the fractional knapsack problem. To determine the most profitable configuration, we will start by computing the ratio of the value to the weight for each item. We can then sort items by their value to weight ration and pull items out of the list and into our knapsack until we reach the weight limit. For the final item in the knapsack, we may need to place a fractional item. This is not the case for the other items, as it would not make sense to include a fraction of a more valuable item until we either run out of space or consume the entire item.

Algorithm 1 Fractional Knapsack

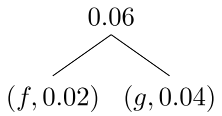
```
function KNAPSACK(weights, values, capacity)
    items ← $(r_i : r_i = values[i]/weights[i])$
    knapsack ← empty set
    while $capacity > 0$ do
        $maximum_i \leftarrow max(items)$
        if $capacity \geq weights[i]$ then
            knapsack.push($values[i]$)
            $capacity \leftarrow capacity - weights[i]$
            remove items[$i$]
            remove weights[$i$]
            remove values[$i$]
        else
            ratio ← $capacity/weight[i]$
            knapsack.push($radio \cdot values[i]$)
            $capacity \leftarrow 0$
        end if
    end while
    return $knapsack$
end function
```

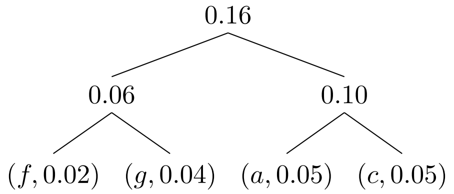
Problem 2

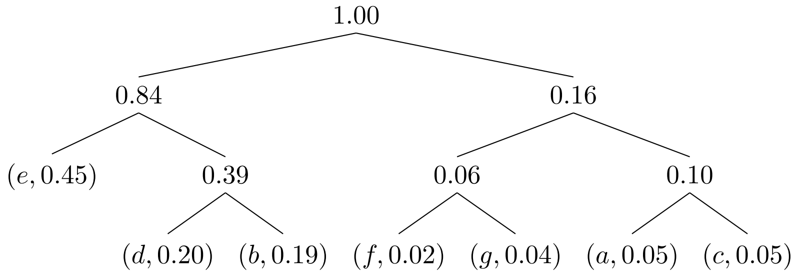
Step	V	MST	Weights
0	$\{1, 2, 3, 4, 5\}$	$\{\}$	$(0, \infty, \infty, \infty, \infty)$
1	$\{2, 3, 4, 5\}$	$\{1\}$	$(0, 11, 4, 2, \infty)$
2	$\{2, 3, 5\}$	$\{1, 4\}$	$(0, 11, 4, 2, 7)$
3	$\{2, 5\}$	$\{1, 4, 3\}$	$(0, 11, 4, 2, 5)$
4	$\{2\}$	$\{1, 4, 3, 5\}$	$(0, 11, 4, 2, 7)$
5	$\{\}$	$\{1, 4, 3, 5, 2\}$	$(0, 11, 4, 2, 7)$



Problem 3







Symbol	Code	Frequency
a	001	0.05
b	100	0.19
c	000	0.05
d	101	0.20
e	11	0.45
f	011	0.02
g	010	0.04

$$\mathcal{L} = 3 \cdot 0.05 + 3 \cdot 0.19 + 3 \cdot 0.05 + 3 \cdot 0.20 + 2 \cdot 0.45 + 3 \cdot 0.02 + 3 \cdot 0.04 = 2.55$$

Problem 4

Round	m1	m2	m3	m4	m5
0	—	—	—	—	—
1	w3	w4	—	w1	w5
2	w3	—	w1	w4	w5
3	w3	—	w1	w4	w5
4	w3	w1	—	w4	w5
5	w3	w1	—	w4	w5
6	w3	w1	w2	w4	w5

Results

M_i	W_i
1	3
2	1
3	2
4	4
5	5