Problem 1

b)

Iteration	x_1	x_2	x_3	x_4	*entering*	*exiting*
initial	1/10	1/100	1/1000	1/10000	w_1	w_2
1.	9/10	9/100	9/1000	9/10000	w_3	w_4
2.	9/10	91/100	91/1000	91/10000	w_2	w_1
3.	1/10	99/100	99/1000	99/10000	w_5	w_6
4.	1/10	99/100	901/1000	820/9101	w_1	w_2
5.	9/10	91/100	909/1000	908/9989	w_4	w_3
6.	9/10	9/100	991/1000	11/111	w_2	w_1
7.	1/10	1/100	999/1000	899/8999	w_7	w_8
8.	1/10	1/100	999/1000	298/331	w_1	w_2
9.	9/10	9/100	991/1000	82/91	w_3	w_4
10.	9/10	91/100	909/1000	792/871	w_2	w_1
11.	1/10	99/100	901/1000	802/901	w_6	w_5
12.	1/10	99/100	99/1000	102/103	w_1	w_2
13.	9/10	91/100	91/1000	334/337	w_4	w_3
14.	9/10	9/100	9/1000	1248/125	w_2	w_1
15.	1/10	1/100	1/1000	1		

It took 15 iterations in Phase 2 to find the optimal solution. Which is equal to $2^n - 1$.

c)

The worst case of the simplex algorithm is exponential as the number of edges between intersection increases exponentially by the number of variables. Since the worst case would be to visit all intersections before finding the optimal solution, the algorithm grows exponentially.

d)

The value of x_1 only changes in the kth pivot when the entering variable is w_1 or w_2. The value of x_1 changes between 1/10 and 1/9 every other pivot.

Additionally, when performing the kth pivot, the sign of the coefficient in the column corresponding to the current pivot and all preceding columns are flipped.

Problem 2

a) Implementation found in planning.py

b)

Objective function (total profit) = 14

Variable	Value
m_0	0
m_1	0
m_2	10
n_0	3
n_1	6

c)

The highest prices we should be willing to pay per unit of capacity expansion for capacity in and capacity out, is 0 and 1.56 respectively.