Brandon Swanson CS 325 Project 3 11/15/15

Problem 1:

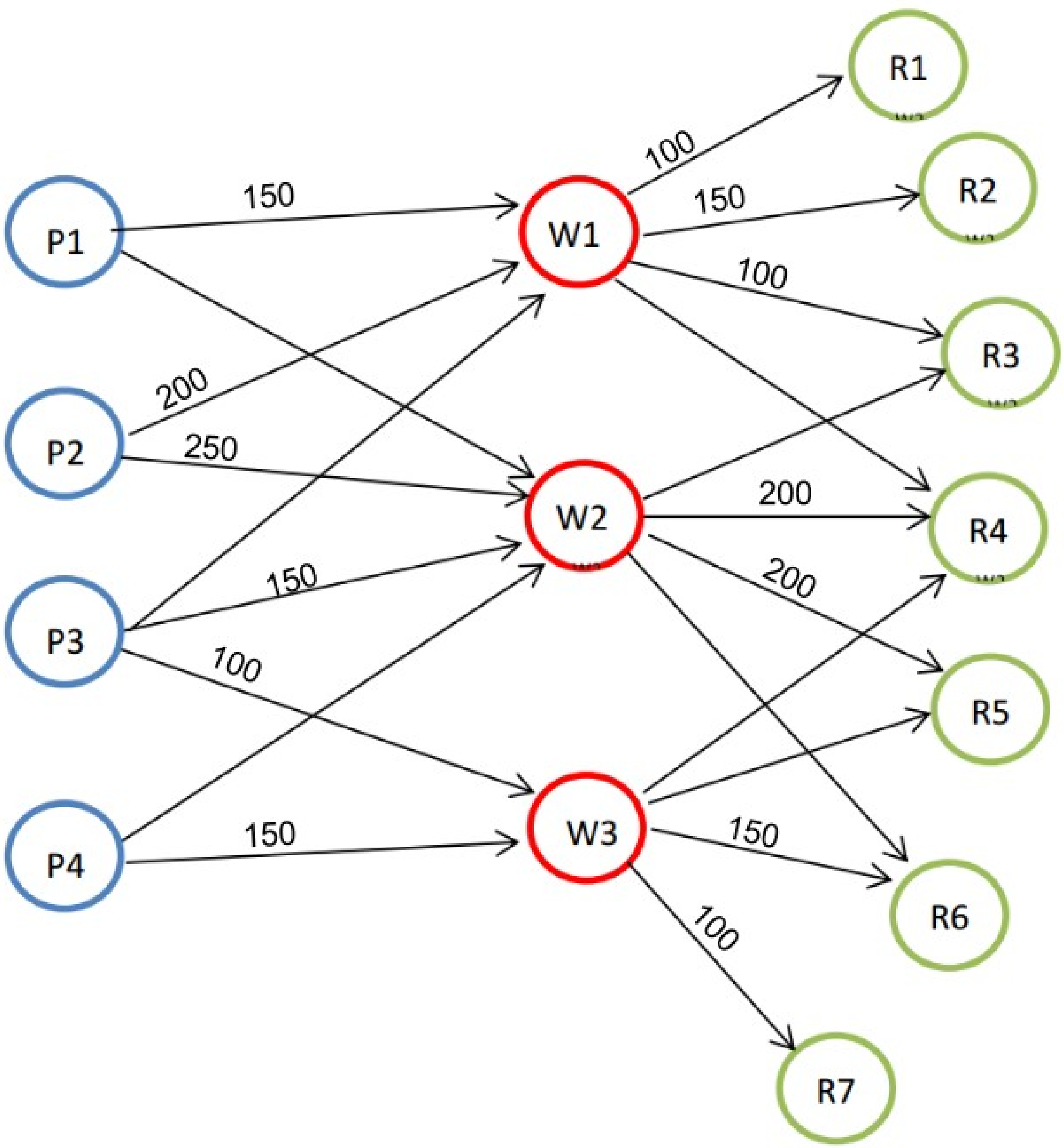
A) Lindo code used to generate solution

```
min 10 p1w1 + 15 p1w2 +
11 p2w1 + 8 p2w2 +
13 p3w1 + 8 p3w2 + 9 p3w3 +
14 p4w2 + 8 p4w3 +

5 w1r1 + 6 w1r2 + 7 w1r3 + 10 w1r4 +
12 w2r3 + 8 w2r4 + 10 w2r5 + 14 w2r6 +
14 w3r4 + 12 w3r5 + 12 w3r6 + 6 w3r7

st
	!supply constraints
	p1w1 + p1w2 <=150
	p2w1 + p2w2 <=450
	p3w1 + p3w2 + p3w3 <=250
	p4w2 + p4w3 <=150

	!warehouse input = output constraint
	w1r1 + w1r2 + w1r3 + w1r4 - p1w1 - p2w1 - p3w1 = 0
	w2r3 + w2r4 + w2r5 + w2r6 - p1w2 - p2w2 - p3w2 - p4w2 = 0
	w3r4 + w3r5 + w3r6 + w3r7 - p3w3 - p4w3 = 0


	!demand constraints
	w1r1 >= 100
	w1r2 >= 150
	w1r3 + w2r3 >= 100
	w1r4 + w2r4 + w3r4 >= 200
	w2r5 + w3r5 >= 200
	w2r6 + w3r6 >= 150
	w3r7 >= 100

!nonegativity constraint
	p1w1 >0
	p1w2 >0
	p2w1 >0
	p2w2 >0
	p3w1 >0
	p3w2 >0
	p3w3 >0
	p4w2 >0
	p4w3 >0
	w1r1 >0
	w1r2 >0
	w1r3 >0
	w1r4 >0
	w2r3 >0
	w2r4 >0
	w2r5 >0
	w2r6 >0
	w3r4 >0
	w3r5 >0
```

```
        w3r6 >0
        w3r7 >0

end
```

Minimum Costs: $17100
Routes:

P1 to W1 :150
P2 to W1 :200 P2 to W2 :250
P3 to W2 :150 P3 to W3 :100
P4 to W3 :150
W1 to R1 :100 W1 to R2 :150 W1 to R3 :100
W2 to R4 :200 W2 to R5 :200
W3 to R6 :150 W3 to R7 :100



B) It is infeasible without warehouse 2, there is no way to meet all the demands of the retailers. R4 is 50 units short.

Lindo code used to verify:

```
min 10 p1w1 + 11 p2w1 + 13 p3w1 + 9 p3w3 + 8 p4w3 +

5 w1r1 + 6 w1r2 + 7 w1r3 + 10 w1r4 +
14 w3r4 + 12 w3r5 + 12 w3r6 + 6 w3r7

st
      !supply constraints
      p1w1 <=150
      p2w1 <=450
      p3w1 + p3w3 <=250
      p4w3 <=150

      !warehouse input = output constraint
      w1r1 + w1r2 + w1r3 + w1r4 - p1w1 - p2w1 - p3w1 = 0
      w3r4 + w3r5 + w3r6 + w3r7 - p3w3 - p4w3 = 0


      !demand constraints
      w1r1 >= 100
      w1r2 >= 150
      w1r3 >= 100
      w1r4 + w3r4 >= 200
      w3r5 >= 200
      w3r6 >= 150
      w3r7 >= 100


      !nonegativity constraint
      p1w1 >0
      p2w1 >0
      p3w1 >0
      p3w3 >0
      p4w3 >0
      w1r1 >0
      w1r2 >0
      w1r3 >0
      w1r4 >0
      w3r4 >0
      w3r5 >0
      w3r6 >0
      w3r7 >0

end
```

To verify the solution I also added just this single constraint to the original problem 1 lindo file:

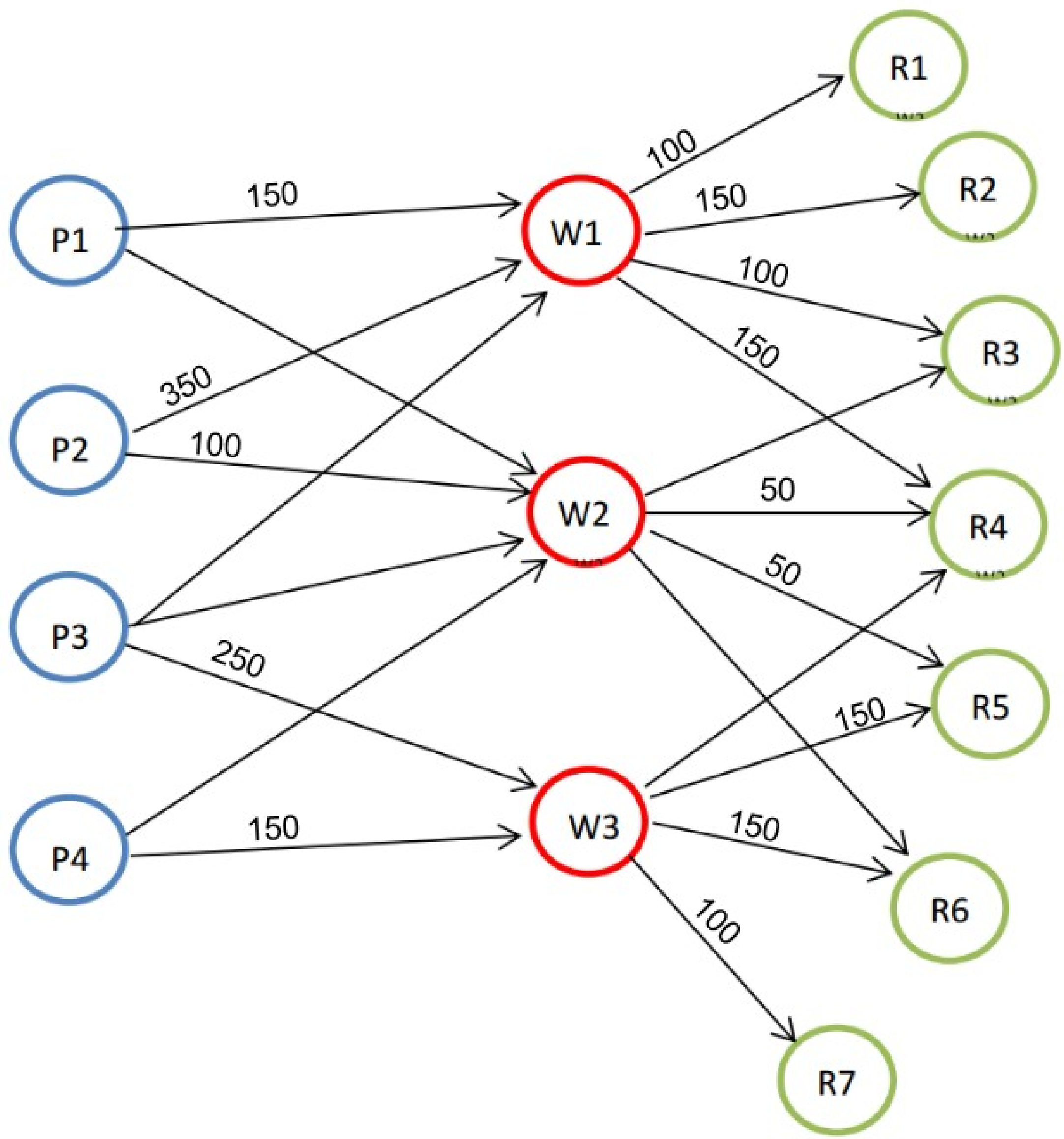
```
      w2r3 + w2r4 + w2r5 + w2r6 + p1w2 + p2w2 + p3w2 + p4w2 = 0
```

C) It is feasible to meet all the demands with the reduced capacity of warehouse 2 with an increased cost of $18,300
to get this solution the the following constraint was added

```
p1w2 + p2w2 + p3w2 + p4w2 = 100
```

It was only necessary to constrain the routes from production to warehouse because there are already constraints in place to ensure input and output of the warehouses are equal.

Minimum Cost: $18,300
Routes:
P1 to W1: 150 P2 to W1: 350 P2 to W2: 100 P3 to W3: 250
P4 to W3: 150 W1 to R1: 100 W1 to R2: 150 W1 to R3: 100
W1 to R4: 150 W2 to R4: 50 W2 to R5: 50 W3 to R5: 150
W3 to R6: 150 W3 to R7: 100



D)

m is number of production sites

n is number of wharehouse sites

o is number of retail sites

pwcost is matrix of cost of shipping from production $_i$ to wharehouse $_j$

wrcost is matrix of cost of shipping from wharehouse$_k$ to retail$_l$

s_i is product available at each production site

d_l is demand at each retail site

c_i is capacity at each wharehouse site

Objective Function:

minimize $\sum_{i=1}^{m} \sum_{j=1}^{n} pwcost_{ij}\, pw_{ij} + \sum_{k=1}^{n} \sum_{l=1}^{o} wrcost_{kl}\, wr_{ki}$

subject to:

$$pw_{ij} \geq 0 ; \forall\, i=1...m, \forall\, j=1...n$$
$$wr_{kl} \geq 0 ; \forall\, k=1...n, \forall\, l=1...o$$
$$\sum_{j=1}^{n} pw_{ij} \leq s_i \forall\, i=1...m$$
$$\sum_{l=1}^{o} wr_{kl} = d_l \forall\, k=1...n$$
$$\sum_{i=1}^{n} pw_{ij} \geq \sum_{l=1}^{o} wr_{il} \forall\, i=1..n$$
$$\sum_{i=1}^{n} pw_{ij} \geq c_i \forall\, i=1..n$$

Problem 2:

A)

LINDO CODE

```
min 21t + 16l + 40s + 41c + 585ss + 120st + 164cp + 884o
st
      PROTIEN).85t + 1.62l + 2.86s + .93c + 23.4ss + 16st + 9cp >= 15

      MT2GRAMFAT).33t + .2l + .39s + .24c + 48.7ss + 5st + 2.6cp + 100o >=2
      LT8GRAMFAT).33t + .2l + .39s + .24c + 48.7ss + 5st + 2.6cp + 100o <=8

      CARBOHYDRAYTES)4.64t + 2.37l + 3.63s + 9.58c + 15ss + 3st + 27cp >=4

      SODIUM)9t + 28l + 65s + 69c + 3.8ss + 120st + 78cp <=200

      PERCENTLEAFY).6s + .6l - .4t - .4c - .4ss - .4st - .4cp - .4o >= 0

      t >= 0
      l >= 0
      s >= 0
      c >= 0
      ss >= 0
      st >= 0
      cp >= 0
      o >= 0

end
```

The minimum calorie salad that meets all of the requirements has 114.7541 calories and contains
58.54801 grams of Lettuce
87.82201 grams of Smoked Tofu
and costs 2.33$

B)

LINDO CODE

```
min t + .75l + .5s + .5c + .45ss + 2.15st + .95cp + 2o
st
      PROTIEN).85t + 1.62l + 2.86s + .93c + 23.4ss + 16st + 9cp >= 15

      MT2GRAMFAT).33t + .2l + .39s + .24c + 48.7ss + 5st + 2.6cp + 100o >=2
      LT8GRAMFAT).33t + .2l + .39s + .24c + 48.7ss + 5st + 2.6cp + 100o <=8

      CARBOHYDRAYTES)4.64t + 2.37l + 3.63s + 9.58c + 15ss + 3st + 27cp >=4

      SODIUM)9t + 28l + 65s + 69c + 3.8ss + 120st + 78cp <=200

      PERCENTLEAFY).6s + .6l - .4t - .4c - .4ss - .4st - .4cp - .4o >= 0

      t >= 0
      l >= 0
      s >= 0
      c >= 0
      ss >= 0
      st >= 0
      cp >= 0
      o >= 0

end
```

The low-cost salad has a cost of 1.55$ and has 278.4884 calories

C)

It seems that the tools for solving multiple optimization problems like this one are usually interactive with the a person being presented with iterative solutions an then inputing information about tradeoffs or pairwise comparisons.   Through this negotiation improved solutions are presented until the most preferred solution is found.

Barring access to such interactive tools for linear programming Veronica can add upper bound constraints on calories and cost and then minimize cost and calories in turn to discover the minumum calories if the cost is allowed to be equal to the upper bound and vice versa for the minimum cost. From here she can use this information to tighten the upper bound on the constraints of cost and calorie to some value between her original upper bound and their absolute minimum, checking if such a solution is feasible.

I would suggest a salad that contains

Ingredient	**Grams**
Tomato	0
Lettuce	0
Spinach	59.28
Carrot	0
Sunflower Seeds	8.84
Smoked Tofu	57.52
Chickpeas	22.56
Oil	0

With a cost of: $1.78 (and profit of $3.22 if sold at 5$)
and total calories: 181.47

I arrived at this combination of ingredients through a system of binary searching for a feasible and nearly optimal solution.
I first added the constraints for the upper bounds of calories and cost as such

```
COST) t + .75l + .5s + .5c + .45ss + 2.15st + .95cp + 2o <=2
CALORIE) 21t + 16l + 40s + 41c + 585ss + 120st + 164cp + 884o <= 249
```

and then solved with the objective function minimizing cost and then calories in turn

If the cost was allowed to be 2 dollars then the optimum salad had only 134.76 calories.
If the calories were allowed to be 249 then the optimum salad cost only $1.62

I then solved once again for both minimizing cost and calorie in turn updating the cost and calorie constraints to be the mid point between their optimum (min) values and the upper bound for each

```
COST) t + .75l + .5s + .5c + .45ss + 2.15st + .95cp + 2o <=1.8125
CALORIE) 21t + 16l + 40s + 41c + 585ss + 120st + 164cp + 884o <= 191.88
```

With these new constraints an optimum calorie salad cost $1.81 and has 171.074 calories
and an optimum cost salad cost $1.76 and 191.88 calories.

Again the constraints were updated to be the midpoint between these new optimums
cost constraint = (1.81 + 1.76) /2   calories constrain = (171.074 + 191.88) / 2

```
COST) t + .75l + .5s + .5c + .45ss + 2.15st + .95cp + 2o <=1.7874775
CALORIE) 21t + 16l + 40s + 41c + 585ss + 120st + 164cp + 884o <= 181.477
```

This was the final iteration because a feasible solution with these constraints was found and the cost and calories objective function caused negligible variation in in the cost and calories and ingredients variables. The two optimizations differed in cost by less than a penny and by a thousandth of a calorie and hundredth of a gram for each ingredient. So this seems like a good middle ground of optimizing both calorie and cost.

Problem 3:

A)

LINDO CODE

```
max a+b+c+d+e+f+g+h+i+j+k+l+m

st

      a=0

      b - a <= 2
      c - a <= 3
      d - a <= 8
      h - a <= 9
      a - b <= 4
      c - b <= 5
      e - b <= 7
      f - b <= 4
      d - c <= 10
      b - c <= 5
      g - c <= 9
      i - c <= 11
      f - c <= 4
      a - d <= 8
      g - d <= 2
      j - d <= 5
      f - d <= 1
      h - e <= 5
      c - e <= 4
      i - e <= 10
      i - f <= 2
      g - f <= 2
      d - g <= 2
      j - g <= 8
      k - g <= 12
      i - h <= 5
      k - h <= 10
      a - i <= 20
      k - i <= 6
      j - i <= 2
      m - i <= 12
      i - j <= 2
      k - j <= 4
      l - j <= 5
      h - k <= 10
      m - k <= 10
      m - l <= 2
end
```

Shortest Possible Distance to all vertices's from A
B:2
C:3
D:8
E:9
F:6
G:8
H:9
I:8
J:10
K:14
L:15
M:17

B)
In this case, that there is a vertex unreachable from the source then the solution will be unbounded

C)
To accomplish this I will reverse the directions of the edges of the graph and solve the shortest path from m to all other vertices's like so:
LINDO CODE
```
max a+b+c+d+e+f+g+h+i+j+k+l+m

st

      m=0

      a - b <= 2
      a - c <= 3
      a - d <= 8
      a - h <= 9
      b - a <= 4
      b - c <= 5
      b - e <= 7
      b - f <= 4
      c - d <= 10
      c - b <= 5
      c - g <= 9
      c - i <= 11
      c - f <= 4
      d - a <= 8
      d - g <= 2
      d - j <= 5
      d - f <= 1
      e - h <= 5
      e - c <= 4
      e - i <= 10
      f - i <= 2
      f - g <= 2
      g - d <= 2
      g - j <= 8
      g - k <= 12
      h - i <= 5
```

```
        h - k <= 10
        i - a <= 20
        i - k <= 6
        i - j <= 2
        i - m <= 12
        j - i <= 2
        j - k <= 4
        j - l <= 5
        k - h <= 10
        k - m <= 10
        l - m <= 2
end
```

Shortest Possible Distances from all vertices's to M
A:17
B:15
C:15
D:12
E:19
F:11
G:14
H:14
I:9
J:7
K:10
L:2

D)

To solve this problem I formulated two linear program similar to parts a and b. One linear program that can find the shortest path from vertex I to all other vertices's, and another with the edge weights reversed that finds the shortest distance from all veracities to vertex I excluding unreachable veracities. These distances can then be combined to form the matrix below. LINDO code on next page.

TO VERTEX

FROM VERTEX

	A	B	C	D	E	F	G	H	J	K
A	28	26	26	23	30	22	25	25	22	35
B	30	28	28	25	32	24	27	27	24	37
C	31	29	29	26	33	25	28	28	25	38
D	36	34	34	31	38	30	33	33	30	43
E	37	35	35	32	39	31	34	34	31	44
F	34	32	32	29	36	28	31	31	28	41
G	36	34	34	31	38	30	33	33	30	43
H	24	22	22	19	26	18	21	21	18	31
J	10	8	8	5	12	4	7	7	4	17
K	14	12	12	9	16	8	11	11	8	21
L	15	13	13	10	17	9	12	12	9	22
M	17	15	15	12	19	11	14	14	11	24

```
max a+b+c+d+e+f++g+h+i+j+k+l+m

st

     i=0

     b - a <= 2
     c - a <= 3
     d - a <= 8
     h - a <= 9
     a - b <= 4
     c - b <= 5
     e - b <= 7
     f - b <= 4
     d - c <= 10
     b - c <= 5
     g - c <= 9
     i - c <= 11
     f - c <= 4
     a - d <= 8
     g - d <= 2
     j - d <= 5
     f - d <= 1
     h - e <= 5
     c - e <= 4
     i - e <= 10
     i - f <= 2
     g - f <= 2
     d - g <= 2
     j - g <= 8
     k - g <= 12
     i - h <= 5
     k - h <= 10
     a - i <= 20
     k - i <= 6
     j - i <= 2
     m - i <= 12
     i - j <= 2
     k - j <= 4
     l - j <= 5
     h - k <= 10
     m - k <= 10
     m - l <= 2
end
```

```
max a+b+c+d+e+f+g+h+i+j+k

st

     i=0

     a - b <= 2
     a - c <= 3
     a - d <= 8
     a - h <= 9
     b - a <= 4
     b - c <= 5
     b - e <= 7
     b - f <= 4
     c - d <= 10
     c - b <= 5
     c - g <= 9
     c - i <= 11
     c - f <= 4
     d - a <= 8
     d - g <= 2
     d - j <= 5
     d - f <= 1
     e - h <= 5
     e - c <= 4
     e - i <= 10
     f - i <= 2
     f - g <= 2
     g - d <= 2
     g - j <= 8
     g - k <= 12
     h - i <= 5
     h - k <= 10
     i - a <= 20
     i - k <= 6
     i - j <= 2
     i - m <= 12
     j - i <= 2
     j - k <= 4
     j - l <= 5
     k - h <= 10
     k - m <= 10
     l - m <= 2
end
```