PROBLEM 1

Standard form given below:

$$\min \quad -x-y$$

$$\text{st.} \quad 2x+y^{+}-y^{-}+s_1=3$$

$$x+3y^{+}-3y^{-}+s_2=5$$

$$x,y^{+},y^{-},s_1,s_2\geq 0$$

PROBLEM 2

1)

We introduce a set of multipliers—one for each constraint and we add up the constraints to obtain:

$$2xz_1 + yz_1 + xz_2 + 3yz_2 \leq 3z_1 + 5z_2$$

$$x(2z_1 + z_2) + y(z_1 + 3z_2) \leq 3z_1 + 5z_2$$

For the left-hand side of this inequality to upper bound the objective function, we need:

$$x + y \leq (2z_1 + z_2)x + (z_1 + 3z_2)y$$

This holds when:

$$2z_1 + z_2 \geq 1$$

$$z_1 + 3z_2 \geq 1$$

$$z_1, z_2 \geq 0$$

Thus, we derive the dual LP:

$$\min \quad 3z_1 + 5z_2$$

$$\text{st.} \quad 2z_1 + z_2 \geq 1$$

$$z_1 + 3z_2 \geq 1$$

$$z_1, z_2 \geq 0$$

2)

Optimal Solution for Primal

At vertex $(0,0)$ objective function $0+0=0$

We have to find the intersection of $2x+y=3$ and $x+3y=5$. From $2x+y=3$ we get $y=3-2x$

we use this value and get from $x+3y=5$:

$$x+3(3-2x)=5 \Rightarrow -5x=-4 \Rightarrow x=\frac{4}{5}$$

$$y=3-2\cdot\frac{4}{5}=3-\frac{8}{5}=\frac{7}{5}$$

At vertex $\left(\frac{4}{5},\frac{7}{5}\right)$ objective function $z=\frac{4}{5}+\frac{7}{5}=\frac{11}{5}=2.2$

For $x=0$:

$$2x+y\leq 3 \Rightarrow y\leq 3$$

$$x+3y\leq 5 \Rightarrow y\leq\frac{5}{3}$$

At vertex $\left(0,\frac{5}{3}\right)$ objective function $z=0+\frac{5}{3}=\frac{5}{3}\approx 1.67$

For $y=0$:

$$2x\leq 3 \Rightarrow x\leq\frac{3}{2}$$

$$x\leq 5$$

At vertex $\left(\frac{3}{2},0\right)$ objective function $\frac{3}{2}+0=\frac{3}{2}=1.5$

For vertex $\left(\frac{4}{5},\frac{7}{5}\right)$ we get the maximum objective function which is $\frac{11}{5}=2.2$

Optimal Solution for Dual

For next vertex , we have to get the intersection of $2z_1 + z_2 = 1$ and $z_1 + 3z_2 = 1$

From the eqn, we get

$$1 - 2z_1 = \frac{1 - z_1}{3} \Rightarrow 3 - 6z_1 = 1 - z_1 \Rightarrow -5z_1 = -2 \Rightarrow z_1 = \frac{2}{5}$$

$$z_2 = 1 - 2\left(\frac{2}{5}\right) = 1 - \frac{4}{5} = \frac{1}{5}$$

At vertex $\left(\frac{2}{5}, \frac{1}{5}\right)$**,**

$$3z_1 + 5z_2 = 3\left(\frac{2}{5}\right) + 5\left(\frac{1}{5}\right) = \frac{6}{5} + \frac{5}{5} = \frac{11}{5} = 2.2$$

If $z_1 = 0$, we get

$$z_2 \geq 1$$

$$z_2 \geq \frac{1}{3}$$

At vertex $(0, 1)$**,**

$$3z_1 + 5z_2 = 3(0) + 5(1) = 5$$

If $z_2 = 0$, we get

$$z_1 \geq \frac{1}{2}$$

$$z_1 \geq 1$$

At vertex $(1, 0)$**,**

$$3z_1 + 5z_2 = 3(1) + 5(0) = 3$$

The minimum objective function value is achieved at $\left(\frac{2}{5}, \frac{1}{5}\right)$ with $\frac{11}{5} = 2.2$.

PROBLEM 3

$$
\begin{aligned}
\max \quad & 3x_1 - 2x_2 - x_4 \\
\text{st.} \quad & x_1 + 3x_3 - 2x_4 \leq 6 \quad (1) \\
& 2x_2 - x_3 + x_4 \leq 4 \quad (2) \\
& x_1 \geq 0 \quad (3) \\
& x_2 \geq 0 \quad (4) \\
& x_3 \geq 0 \quad (5) \\
& x_4 \geq 0 \quad (6)
\end{aligned}
$$

For first iteration we are choosing x_1, because it is the only positive coefficient in objective function. Releasing constraint (3), constraint (1) becomes tight at $x_1 = 6$. The new vertex $(1), (4), (5), (6)$ has local coordinates (y_1, y_2, y_3, y_4). So, we get $y_1 = 6 + 2x_4 - 3x_3 - x_1, y_2 = x_2, y_3 = x_3, y_4 = x_4$. Replacing value we get $x_1 = 6 - y_1 - 3y_3 + 2y_4$. So the LP,

$$
\begin{aligned}
\max \quad & 18 - 3y_1 - 2y_2 - 9y_3 + 5y_4 \\
\text{st.} \quad & y_1 \geq 0 \quad (1) \\
& 2y_2 - y_3 + y_4 \leq 4 \quad (2) \\
& y_1 + 3y_3 - 2y_4 \leq 6 \quad (3) \\
& y_2 \geq 0 \quad (4) \\
& y_3 \geq 0 \quad (5) \\
& y_4 \geq 0 \quad (6)
\end{aligned}
$$

For 2nd iteration, we are choosing y_4 because it is the only positive coefficient. Releasing constraint (6), constraint (2) becomes tight at $y_4 = 4$. The new vertex $\{(1), (4), (5), (2)\}$ has local coordinates (z_1, z_2, z_3, z_4). So$z_1 = y_1, z_2 = y_2, z_3 = y_3, z_4 = 4 - 2y_2 + y_3 - y_4$By replacing , we get $y_4 = 4 - 2z_2 + z_3 - z_4$. So , the new LP

$$\begin{aligned} \text{maximize} \quad & 38 - 3z_1 - 12z_2 - 4z_3 - 5z_4 \\ \text{subject to} \quad & z_1 \geq 0 \quad (1) \\ & z_4 \geq 0 \quad (2) \\ & z_1 + 4z_2 + z_3 + 2z_4 \leq 14 \quad (3) \\ & z_2 \geq 0 \quad (4) \\ & z_3 \geq 0 \quad (5) \\ & 2z_2 - z_3 + z_4 \leq 4 \quad (6) \end{aligned}$$

For the 3rd iteration, all coefficients are negative, so we reached the optimum where$z_1 = 0, z_2 = 0, z_3 = 0, z_4 = 0,$ and optimal value $= 38$. Replacing y variables, we get $y_1 = 0, y_2 = 0, y_3 = 0, y_4 = 4$. Replacing x variables, we get $x_1 = 6 - 0 - 0 + 8 = 14, x_2 = 0, x_3 = 0, x_4 = 4$.

So, the optimal solution is,

$$x_1 = 14, \quad x_2 = 0, \quad x_3 = 0, \quad x_4 = 4, \quad \text{and optimal value } = 38$$

PROBLEM 4

1

Let each vertex v is represented by a variable x_v. $x_v = 1$ if the v-th vertex is included in the vertex cover, and $x_u = 0$ otherwise. Our goal is to minimize the number of vertices selected in the vertex cover set.

The Linear Program is,

$$\min \quad \sum_{v \in V} x_v$$

$$\text{st.} \quad x_v + x_w \geq 1, \quad \forall \{v, w\} \in E$$

$$x_v \in \{0, 1\}, \quad \forall v \in V$$

2

Let a triangle graph $G = (V, E)$ .It has three vertices $V = \{a, b, c\}$ and edges $E = \{(a, b), (b, c), (c, a)\}$. The Linear Program is:

$$\min \quad x_a + x_b + x_c$$

$$\text{st.} \quad x_a + x_b \geq 1$$

$$x_b + x_c \geq 1$$

$$x_c + x_a \geq 1$$

$$0 \leq x_a, x_b, x_c \leq 1$$

If we set, $x_a = x_b = x_c = \frac{1}{2}$, it satisfies the constraints and as well minimizes $x_a + x_b + x_c = \frac{3}{2}$.This fractional solution does not correspond to an actual vertex cover as vertex cover values must be integers.The minimum vertex cover for this graph is any two of the three vertices, e.g., $\{a, b\}$, $\{b, c\}$, or $\{c, a\}$, each with size 2.

$$\text{Integrality Gap} = \frac{\text{size of minimum integral vertex cover}}{\text{optimal LP solution}} = \frac{2}{\frac{3}{2}} = \frac{4}{3}$$

So, for some graphs, vertex cover determined by LP is not optimal

PROBLEM 5

Suppose the units are x_1, x_2 and x_3. We have to minimize $0.3^{x_1}.0.4^{x_2}.0.3^{x_3}$. We can make this linear by logarithm.

$$x_1 \cdot \log(0.3) + x_2 \cdot \log(0.4) + x_3 \cdot \log(0.2)$$

Each compartment have a probability of no more than 0.05 that all its units fail. So we get,

$$0.3^{x_1} \leq 0.05 \Rightarrow x_1 \geq \frac{\log(0.05)}{\log(0.3)} \Rightarrow x_1 \geq 2.48$$

$$0.4^{x_2} \leq 0.05 \Rightarrow x_2 \geq \frac{\log(0.05)}{\log(0.4)} \Rightarrow x_1 \geq 3.27$$

$$0.2^{x_3} \leq 0.05 \Rightarrow x_3 \geq \frac{\log(0.05)}{\log(0.4)} \Rightarrow x_1 \geq 1.86$$

The LP is,

$$\min \quad x_1 \cdot \log(0.3) + x_2 \cdot \log(0.4) + x_3 \cdot \log(0.2)$$

$$\text{st.} \quad 40x_1 + 50x_2 + 30x_3 \leq 500$$

$$15x_1 + 20x_2 + 10x_3 \leq 200$$

$$30000x_1 + 35000x_2 + 25000x_3 \leq 400000$$

$$x_1 \geq 3$$

$$x_2 \geq 4$$

$$x_3 \geq 2$$

$$x_1, x_2, x_3 \in \mathbb{Z}$$

PROBLEM 6

Variable: $f_e^{(i)}$ is the flow on edge e for the i-th flow from source s_i to sink t_i. Each edge has total k variable, making total $E.K$ variables.

Objective Function: our objective is to maximize the sum of flows across all source-sink pairs

$$\max \quad \sum_{i=1}^{k} \sum_{\text{edge } e \text{ out of } s_i} f_e^{(i)}$$

Constraints:

For each $i = 1, \ldots, k$ excluding source and sink the flow into node q must equal the flow out of q:

$$\sum_{e:(p,q)\in E} f_{(p,q)}^{(i)} = \sum_{e:(q,r)\in E} f_{(q,r)}^{(i)}$$

For each edge $e \in E$, the total flow across all flows $f^{(i)}$ cannot be more than the edge's capacity c_e:

$$\sum_{i=1}^{k} f_e^{(i)} \leq c_e$$

For each source-sink pair (s_i, t_i), the flow out of s_i must be at least the demand d_i

$$\sum_{\text{edge } e \text{ out of } s_i} f_j^i \geq d_i, \quad 1 \leq i \leq k$$

All flows must be non-negative.

$$f_e^{(i)} \geq 0 \quad \forall e \in E,\ i = 1, \ldots, k$$

When LP terminates, we can get the flow for each pair.

$$f^{(i)} = \sum_{\text{edge } e \text{ out of } s_i} f_e^i \quad 1 \leq i \leq k$$

PROBLEM 7

For any problem $\Pi \in$ NP, there exists a polynomial-time verification algorithm which run for $O(p_i(n))$. For any input of size n, the size of the certificate is at most $p_j(n)$. $p_i(n)$ and $p_j(n)$ are polynomial in n.

To solve the problem, we can enumerate all possible certificates of length at most $p_j(n)$ and for each certificate we can run the verification algorithm to check if the certificate is a valid solution. The number of possible certificates is bounded by $2^{p_i(n)}$ because each bit in the certificate has 2 possible values (0 or 1). we can verify whether this is a valid solution or not using the polynomial-time verification algorithm .Since it runs in polynomial time, we can calculate.

$$O(2^{p_i(n)} \cdot p_j(n)) = O(2^{p_i(n)} \cdot 2^{p_j(n)}) = O(2^{p_i(n)+p_j(n)}) = O(2^{p(n)})$$

So, there is an algorithm which solves THIS in time $O(2^{p(n)})$.

PROBLEM 8

4SAT problem is in NP. For a given truth assignment we have to check whether at least one literal in the clause is true. Since each clause has exactly 4 literals, this check can be done in constant time. Repeat this for all clauses. The total time to verify the truth assignment O(clauses), which is polynomial in the size of the input.

We need to reduce 3SAT problem to 4SAT problem to state that 4SAT is NP hard. We will union one dummy variable to every clause. Suppose the dummy variable is z.It will ensure that every clause has exactly 4 literals.

$$(clause1 \cup z) \cap (clause2 \cup z)........$$

We do not want z to affect the original clause, so z needs to be 0 for the formula to be satisfiable. To ensure that the problem is only satisfiable when z=0, we add 8 more clauses to the intersection of the original clauses.

$$(\bar{z} \cup z_1 \cup z_2 \cup z_3) \cap (\bar{z} \cup z_1 \cup z_2 \cup \bar{z_3}) \cap .... \cap (\bar{z} \cup \bar{z_1} \cup \bar{z_2} \cup \bar{z_3})$$

There are 8 clauses where 1st literal is $\bar{z}$ and other 3 literals are different combination of z_1, $\bar{z_1}$ z_2, $\bar{z_2}$, z_3 , $\bar{z_3}$ For any input, one of these clauses has a combination of z_1, $\bar{z_1}$ z_2, $\bar{z_2}$, z_3 , $\bar{z_3}$ that is equal to 0. So in order to make that clause true, $\bar{z}$ must have to be 1. So z must have to be 0. In this way we can reduce 3SAT to 4SAT problem. So 4SAT is NP Hard.

4SAT is NP and NP Hard. Therefore, this is NP Complete.

PROBLEM 9

We need to show that DOMINATING problem is in NP and NP Hard

First, to show that DOMINATING SET is in NP, we can verify whether a given set of vertices is a dominating set in polynomial time. For every node in the graph, we check whether it is either in the set or adjacent to a node in the set. This verification can be done in polynomial time, so DOMINATING SET in NP

Now, to prove DOMINATING set in NP hard, we reduce a NP hard problem to DOMINATING set. We can choose Vertex Cover problem for the reduction. Let $G(V, E)$ is a graph that has a vertex cover of size k. We need to convert this DOMINATING set instance such that if DOMINATING set has size k then vertex cover has also size k . For the reduction we create a new graph G'. We will not add any isolated vertex from G to G'. All other vertices and edges are added. For every edge e_{uv} in G, we create a new vertex v_{uv} in G'. Besides, we create 2 edges from the new node v_{uv} to the endpoint of original edge e_{uv} . So, there is an edge between v_{uv} to u and v_{uv} to v. No isolated vertices are in G', every node of G' has an edge. if a vertex is adjacent to another vertex. Then there is an edge in G and at least one of them are in the vertex cover. It also satisfies dominating set requirements. For the new vertex v_{uv}, there is an edge in G, u or v in vertex cover. So, v_{uv} satisfies the dominating set requirements. Now if the dominating set has any new node, then we can replace it with u or v. Thus, DOMINATING set consists only of nodes that have edges in the original graph. If there is K sized dominating set, then there is k size vertex cover, making dominating set problem NP hard.

Thus DOMINATING Set problem is NP Complete.

PROBLEM 10

We give the graph G as input to our algorithm. The algorithm calls the procedure, and if there is a Hamilton path it returns s and t, which are the first and last vertices of the Hamiltonian path.

Next, we create a new graph G' by removing all incoming edges to s, and deleting vertex t along with its associated edges. Also if there are multiple edges between nodes, we have to keep 1 edge and remove all other edges. We then call the procedure on the new graph G'. The procedure will return a new path with s and t' as the first and last vertices. Since t has been removed from the graph, the procedure will find a new vertex t', which is the immediate predecessor of t in the original Hamiltonian path.We have removed all incoming edges to s, ensuring that the path always starts with s.

In the next step, we remove t' from the graph and repeat the process. In each iteration, we will identify the immediate predecessor of the previously removed vertex.

The iterations will continue until we have identified all vertices, which will take $v - 1$ iterations. In each iteration, we call the procedure, which runs in polynomial time. Therefore, our algorithm runs in polynomial time overall.

PROBLEM 11

1 We need to show that 3 Color can be verified in polynomial time. We have to pick every edge and have to check the endpoints having different color or not. This can be done in polynomial time , so 3 color in NP.

2) We have to construct a triangle with three vertices. T,F, E, represents True, False, and Excluded colors. These vertices are fully connected and forming a 3-clique. It ensures that each vertex must have a unique color in any valid 3-coloring of the graph. For each boolean variable x_i create two nodes x_i and $\neg x_i$. Create an edge between two to make sure that these two nodes receive different colors in any 3-coloring. Connect each of nodes to E node in the triangle. This connection prevents both x_i and $\neg x_i$ from being colored E . It ensures that one of them must be colored T and other F.

3)

(i) The nodes T,F, E are connected to each other as they have edges between them. So, They have different colors. x_1, x_2, $\bar{x}_3$ are connected to E. They can not be colored as E. They should be colored with either T or F.

(ii) If all of $\{x_1, x_2, \bar{x}_3\}$ are colored F, then all of $\{a_1, a_2, \bar{a}_3\}$ must be colored E, since they each have edges connected to T. b_1 must be colored F as it is connected to T and a_1, which is colored E. b_3 should be colored T because it is connected to F and a_3, which is colored E. There is no valid color for b_2 as it is connected to b_1 which is colored F, b_3 which is colored T, and a_2 which is colored E. So if all of $\{x_1, x_2, \bar{x}_3\}$ are colored F, there is no valid 3-coloring.

(iii) Valid coloring for $\{a_1, a_2, \bar{a}_3\}$ and $\{b_1, b_2, \bar{b}_3\}$ is shown below:

$\{x_1, x_2, \bar{x}_3\}$	$\{a_1, a_2, \bar{a}_3\}$	$\{b_1, b_2, \bar{b}_3\}$
T, T, T	E, F, F	F, E, T
T, T, F	F, F, E	E, F, T
T, F, T	F, E, F	E, F, T
T, F, F	F, E, E	E, F, T
F, T, T	E, F, F	F, T, E
F, T, F	E, F, E	F, E, T
F, F, T	E, E, F	F, F, T

4) We are taking the graph from no 2 here. For each clause c_i ($1 \leq i \leq m$) We are going to add gadgets . Every clause has 3 terms t_{i1}, t_{i2}, t_{i3}. They are either variables or negations. We bring 6 new nodes $\{a_{i1}, a_{i2}, a_{i3}\}$ and $\{b_{i1}, b_{i2}, b_{i3}\}$ for each clause c_i. In refer to 2 question, every terms are connected to E node. We have to connect each of the terms t_{i1}, t_{i2}, t_{i3} to these nodes $\{a_{i1}, a_{i2}, a_{i3}\}$ respectively. Besides, connect each of $\{a_{i1}, a_{i2}, a_{i3}\}$ nodes to True node. Connect $\{a_{i1}, a_{i2}, a_{i3}\}$ to $\{b_{i1}, b_{i2}, b_{i3}\}$ respectively. Connect b_{i1} to T node, b_{i3} to F node, and b_{i2} to both b_{i1} and b_{i3}. When all the gadgets are being added for all clauses, We will get the final graph. we set the variables colored T as True, and the rest as False. If and only if the 3SAT instance is satisfiable, The graph is 3 Colorable.

5) Correctness: For any clause c_i, we considered total 12 nodes. Gadget can be colored if one of the terms are colored T, otherwise they can not colored. If all terms are colored false, then gadget is not colorable. If the graph is 3-colorable, at least one term is colored as T. As each clause has at least one true term, the entire 3SAT is satisfiable. On the other hand, if the 3SAT is satisfiable, then at least one term in each clause must be true. So each gadget is colorable, making entire graph is colorable.

Running Time: We added $2n + 3$ nodes in the graph . For each clause, we added constant number of edges and modes. So overall time complexity is $O(m + n)$. Running time is polynomial.