Question – 1 Solution:

a) General Integer Linear Programming for Minimum Vertex Cover

- A decision variable x_a is taken ∀ a ∈ Vertices V
- x_a = 1 -> we place the 'a' vertex into the cover
  x_a = 0 -> we don't place the 'a' vertex into the cover
- We used Linear cost function and linear constraints on the variables x_a
  To formulate Weighted Vertex Cover Problem

  Weights (W) = ($w_1, w_2, \ldots, w_a$) are the weights for the vertices.
- Minimize the total weights of the vertices in the Cover which subject to the vertices in C form a cover
- Total Weight = $\Sigma_{a \in C}$ weight (V)
  = $\Sigma_{a \in V}$ weight (V) * x_a

  Instead of taking the sum over vertices in the cover, we can take the formulated variable weight multiplied by the vertex x_a
- If a is in cover x_a = 1 else x_a = 0
- Constraints:

  Vertices form a cover
  a ∈ C or b ∈ C ∀ (a, b) ∈ Edges E
  $x_a + x_b >= 1$ ∀ (a, b) ∈ Edges E
  $x_a \in \{0,1\}$ ∀ a ∈ Vertices V

Minimize $\Sigma_{a \in V}$ weight (V) * x_a subject to $x_a + x_b >= 1$ ∀ (a, b) ∈ Edges E and $x_a \in \{0,1\}$ ∀ a ∈ Vertices V

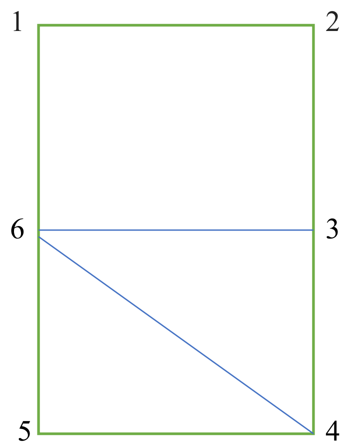
Algorithm:

Begin: Weighted_Vertex_Cover_LP(G)
1) Generate the linear programming corresponding to graph G.
2) Minimize $\Sigma_{a \in V}$ weight (V) * x_a subject to $x_a + x_b >= 1$ ∀ (a, b) ∈ Edges E and $x_a \in \{0,1\}$ ∀ a ∈ Vertices V
3) Solve the 0/1 Linear programming
4) Cover C <- {a ∈ Vertices V: x_a = 1}
5) Return Cover C

End

0/1 Linear Programming is NP-Hard (not a feasible solution)

b) Integer Linear Programming for the given graph



1) Objective: Minimize $\Sigma_{a \in V}$ weight (V) * x_a

   $1 * x_1 + 2 * x_2 + 3 * x_3 + 4 * x_4 + 5 * x_5 + 6 * x_6$

2) Subject to:

$$
\begin{aligned}
x_1 + x_2 &>= 1 \\
x_2 + x_3 &>= 1 \\
x_3 + x_4 &>= 1 \\
x_3 + x_6 &>= 1 \\
x_4 + x_6 &>= 1 \\
x_4 + x_5 &>= 1 \\
x_1 + x_6 &>= 1 \\
x_5 + x_6 &>= 1
\end{aligned}
$$

Such that $x_2 = x_4 = x_6 = 1$

$x_1 = x_3 = x_5 = 0$

the above can be written in matrix form as where variables = vertices = columns in A and constraints = edges = rows in A

AX >= b

$$
A = \begin{pmatrix} 1\,1\,0\,0\,0\,0 \\ 0\,1\,1\,0\,0\,0 \\ 0\,0\,1\,1\,0\,0 \\ 0\,0\,1\,0\,0\,1 \\ 0\,0\,0\,1\,0\,1 \\ 0\,0\,0\,1\,1\,0 \\ 1\,0\,0\,0\,0\,1 \\ 0\,0\,0\,0\,1\,1 \end{pmatrix}
\quad X = \begin{pmatrix} x1 \\ x2 \\ x3 \\ x4 \\ x5 \\ x6 \end{pmatrix}
\quad B = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}
$$

3) Considering the greedy approach, we get $x_1 = x_3 = x_5 = x_4 = 1$ and $x_2 = x_6 = 0$ so the sub-optimal cost is 13 whereas optimal $x_2 = x_4 = x_6 = 1$ and $x_1 = x_3 = x_5 = 0$ cost is 12

Sub-optimal value <= 2 * optimal value
13 <= 2*12
13 <= 24

Question 2 Solution:

a. General ILP for Minimum Dominating Set

- a decision variable x_a, $\forall a \in V$ (vertices)
    - **a)** $x_a = 1$ => we place a in the cover
    - **b)** $x_a = 0$ => we don't place a in cover
- We then formulate the weighted vertex cover using cost functions and linear constraints on the variables x_a as Weights (W) = (w_1, w_2, ……., w_a)
- Minimize total weight of vertices in V subject to vertices in D form a dominant set of total weight = $\Sigma_{a\in D}\, d_a$
- Instead of taking the weight for every vertex we take the formulated variable weight multiplied by vertex x_a (if it is in x_a we take $x_a = 1$ and 0 otherwise)
- Minimize $\Sigma_{a\in D}\, W_a \,.\, x_a$, subject to the vertices form a cover if
    - **c)** $a \in$ Dominant Set or all the adjacent vertices in Dominant set , $\forall$ (a, b) $\in$ E
    - **d)** Implies that $x_a = 1$, $x_a \in$ Dominant Set, $x_a + \Sigma_{j\in N(a)}\, x_j \geq 1$, N = {Neighboring vertices of i}
    - **e)** $X_a + \Sigma_{j\in N(a)}\, x_j \geq 1$, $\forall$ neighboring edges $\in$ E → this condition is to ensure the domination
- $X_a \in \{0, 1\}$, $\forall a \in V$

Algorithm:

Begin: Weighted_Dominating_Set_LP(G)

1. while there exist, non-dominated vertices do

    randomly select a vertex a from all non-dominated vertices;

    select a vertex $u \in N[a]$ with the highest degree.

    W := W ∪ {a};

2. Generate LP corresponding to G

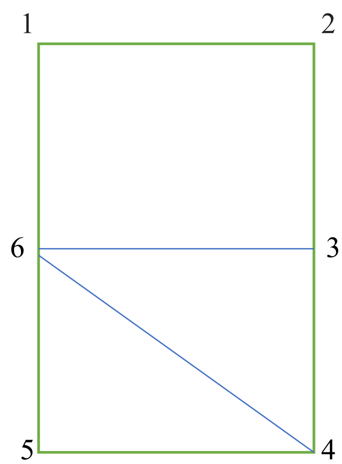
    - Minimize $\Sigma_{a\in D}\, W_a \,.\, x_a$

- Subject to $x_a + \Sigma_{j\in N(a)} x_j \geq 1$, N = {Neighboring vertices of a}
- $x_a \in \{0, 1\}$, $\forall$ a $\in$ V -> this is not a linear constraint and that's why it is called as integer linear programming

3. Solve the linear programming problem
4. Dominating Set ← {a ∈ V: $x_a = 1$}
5. return Dominating Set

End

Given graph:



Objective: We want to minimize the number of vertices in the dominating set. So, the objective is to minimize $1 . x_1 + 2 . x_2 + 3 . x_3 + 4 . x_4 + 5 . x_5 + 6 . x_6$

Each vertex needs to be in the set or have a neighbor in the set. So, we get a constraint for each vertex of the form $\Sigma_{a\in N(v)} x_a \geq 1$, for all v ∈ V.

Subject to

$x_1 + x_2 + x_6 >= 1$, $x_1 + x_2 + x_3 >= 1$, $x_2 + x_3 + x_4 + x_6 >= 1$, $x_3 + x_4 + x_5 + x_6 >= 1$, $x_4 + x_5 + x_6 >= 1$, $x_1 + x_3 + x_4 + x_5 + x_6 >= 1$

$x_a \in \{0, 1\}$

if we represent this as a matrix form, we get AX >= B

$$A = \begin{pmatrix} 1\,1\,0\,0\,0\,1 \\ 1\,1\,1\,0\,0\,0 \\ 0\,1\,1\,1\,0\,1 \\ 0\,0\,1\,1\,1\,1 \\ 0\,0\,0\,1\,1\,1 \\ 1\,0\,1\,1\,1\,1 \end{pmatrix} X = \begin{pmatrix} x1 \\ x2 \\ x3 \\ x4 \\ x5 \\ x6 \end{pmatrix} B = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

Solving the Dominating set problem, we get {1, 6}

thus, we have the integer constraints as $x_1 = x_6 = 1$ and $x_2 = x_3 = x_4 = x_5 = 0$

Considering the greedy approach (starting with the vertex having minimal weight), we get $x_1 = x_4 = 1$ and $x_2 = x_3 = x_5 = x_6 = 0$ so, optimal cost is 5.