**Q1 Solution:**

1. **General ILP for minimum vertex cover:**
   1. Introduce a decision variable xi, ∀ i ∈ V (vertices)
      1. xi = 1 => we put i in the cover
      2. xi = 0 => we don’t put i in cover
   2. We then formulate the weighted vertex cover using cost functions and linear constraints on the variables xi as C = (c1, c2, c3,…., cv)
   3. Minimize total weight of vertices in V subject to vertices in C form a cover of total weight = Σv∈C Cv
   4. Instead of taking the weight for every vertex we take the formulated variable weight multiplied by vertex xi (if it is in xi we take xi = 1 and 0 otherwise)
   5. Minimize Σv∈V Cv . xv,subject to the vertices form a cover if
      1. u ∈ Cover or v ∈ Cover ∀ (u, v) ∈ E
      2. Implies that xu = 1 or xv = 1, ∀ (u, v) ∈ E
      3. xu + xv >= 1, ∀ (u, v) ∈ E
      4. xi ∈ {0, 1}, ∀ i ∈ V

**Algorithm:**

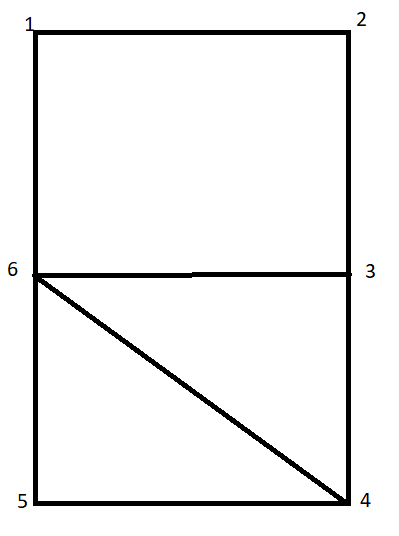
**Begin:** Weighted\_Vertex\_Cover\_LP(G)

1. Generate LP corresponding to G
   * + Minimize Σv∈V Cv . xv
     + Subject to xu + xv >= 1, ∀ (u, v) ∈ E
     + xi ∈ {0, 1}, ∀ i ∈ V -> this is not a linear constraint and that’s why it is called as integer linear programming
2. Solve the linear programming problem
3. Cover 🡨 {v ∈ V: xv = 1}
4. return Cover

**End**

**Note:** This is not a Linear programming problem, it is a 0/1 Linear problem. Unfortunately, 0/1 Linear problem is Np-hard (Solution isn’t feasible)

**Given graph:**



1. Objective is to Minimize 1 . x1 + 2 . x2 + 3 . x3 + 4 . x4 + 5 . x5 + 6 . x6

Subject to x1 + x2 >= 1, x1 + x6 >= 1, x2 + x3 >= 1, x3 + x6 >= 1, x4 + x6 >= 1, x5 + x6 >= 1, x4 + x5 >= 1, x3 + x4 >= 1

We must add 0/1 constraints for the variables to state that the values come from 0 or 1. So, xi ∈ {0, 1}, ∀ i ∈ V is the additional constraint

If we represent this as the matrix form, we get AX >= B where,

A = X = B =

Solving the vertex cover problem, we get the optimal minimum vertex cover

set as {2, 4, 6}

thus, we have the integer constraints as x2 = x4 = x6 = 1

and x1 = x3 = x5 = 0

Considering the greedy approach (starting with the vertex having minimal weight), we get x1 = x3 = x5 = x4 = 1 and x2 = x6 = 0 so, the sub-optimal cost is 13 whereas optimal x2 = x4 = x6 = 1 and x1 = x3 = x5 = 0 cost is 12

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

**Q2 Solution:**

1. **General ILP for Minimum Dominating Set**
   1. Introduce a decision variable xi, ∀ i ∈ V (vertices)
      1. xi = 1 => we put i in the cover
      2. xi = 0 => we don’t put i in cover
   2. We then formulate the weighted vertex cover using cost functions and linear constraints on the variables xi as D = (d1, d2, d3,…., dv)
   3. Minimize total weight of vertices in V subject to vertices in D form a dominant set of total weight = Σv∈D dv
   4. Instead of taking the weight for every vertex we take the formulated variable weight multiplied by vertex xi (if it is in xi we take xi = 1 and 0 otherwise)
   5. Minimize Σv∈D dv . xv,subject to the vertices form a cover if
      1. u ∈ Dominant Set or all the adjacent vertices in Dominant set , ∀ (u, v) ∈ E
      2. Implies that xu = 1, xu ∈ Dominant Set, xi + Σj∈N(i) xj ≥ 1, N = {Neighboring vertices of i}
      3. xi + Σj∈N(i) xj ≥ 1, ∀ neighboring edges ∈ E 🡪 this condition is to ensure the domination
   6. xi ∈ {0, 1}, ∀ i ∈ V

**Algorithm:**

**Begin:** Weighted\_Dominating\_Set\_LP(G)

1. while there exist, non-dominated vertices do

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

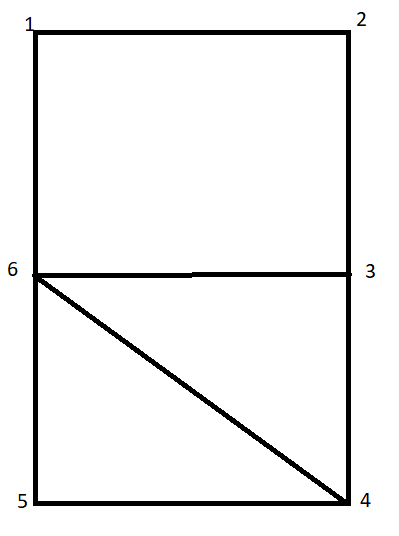
select a vertex u ∈ N[v] with the highest degree.

D := D ∪ {u};

1. Generate LP corresponding to G
   * + Minimize Σv∈D dv . xv
     + Subject to xi + Σj∈N(i) xj ≥ 1, N = {Neighboring vertices of i}
     + xi ∈ {0, 1}, ∀ i ∈ V -> this is not a linear constraint and that’s why it is called as integer linear programming
2. Solve the linear programming problem
3. Dominating Set 🡨 {v ∈ V: xv = 1}
4. 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 . x1 + 2 . x2 + 3 . x3 + 4 . x4 + 5 . x5 + 6 . x6

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 Σi∈N(v) xi ≥ 1, for all v ∈ V.

Subject to

x1 + x2 + x6 >= 1, x1 + x2 + x3 >= 1, x2 + x3 + x4 + x6 >= 1, x3 + x4 + x5 + x6 >= 1, x4 + x5 + x6 >= 1, x1 + x3 + x4 + x5 + x6 >= 1

xi ∈ {0, 1}

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

A = X = B =

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

thus, we have the integer constraints as x1 = x6 = 1 and x2 = x3 = x4 = x5 = 0

Considering the greedy approach (starting with the vertex having minimal weight), we get x1 = x4 = 1 and x2 = x3 = x5 = x6 = 0 so, the optimal cost is 5.