#1. Answer the following questions, provide explanations to your answers:

(1) Graph G has 8 pairwise adjacent vertices. Minimum proper coloring of G uses at least 8 colors Yes No Impossible to say

-Yes, because it is denoting that it is a complete graph so for any complete graph the number of colors required is the number of nodes.

(2) Checking, if two given vertices i and j of a graph G are adjacent is faster using adjacency lists than using adjacency matrix Yes No Don't know

-No, because an adjacency list has all nodes connected to it so it will take more time compared to a matrix which can be directly accessed since nodes are given.

(3) Every tree is a bipartite graph Yes No Don't know

-Yes, because if we take vertex v0 and put it in set ***A*** then take any vertex in the tree that is an even number of edges away and put it in ***A***, then put the rest in ***B.*** If two vertice v1, v2 in ***A*** are connected by edge e then we can make a loop with the path from v0 to v1, v0 to v2, and edge ***e***. Then the original set we were pulling the vertices from is not a tree. By doing the prior process for a tree results in no 2 vertices in ***A*** being connected. Similarly, this holds for ***B***. Then we have shown that a tree is bipartite.

#2. Answer with explanations. Graph G with 10 vertices has 4 pair-wise nonadjacent vertices. Minimum vertex cover of G has

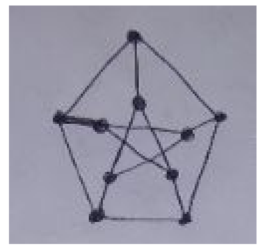
a) at least 4 vertices Yes No Don't know

because____

-Yes, beacuse it has at least 4 vertices because it has 4 pair-wise nonadjacent vertices.

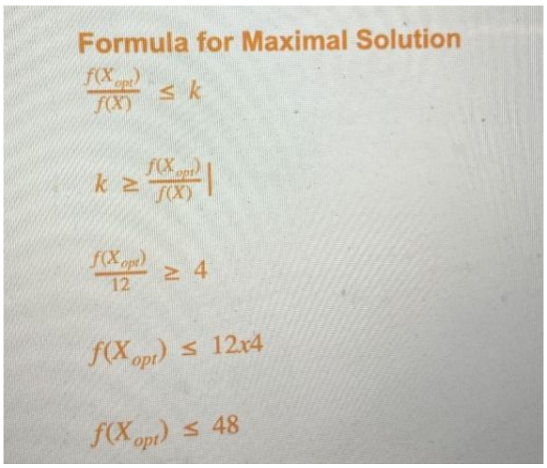
b) at most 6 vertices Yes No Don't know

because____

-Yes, because the graph would look like 

#3. Suppose you have a maximization problem and an algorithm A, that has an approximation ratio of 4. When run on some input I, A produced a solution with cost 12. What can you say about the true (optimal) answer OPT? Explain your chosen answer(s).

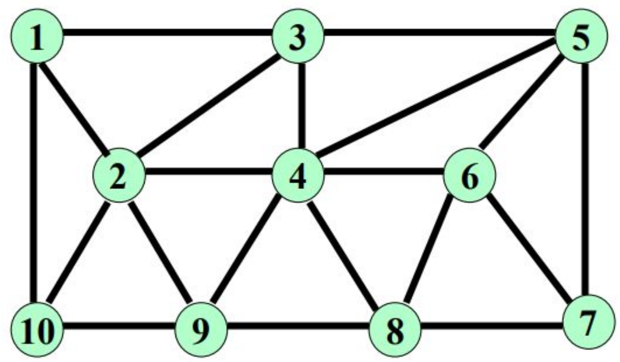
- OP T ≥ 3
- OP T ≤ 3
- OP T ≥ 12
- OP T ≤ 12
- OP T ≥ 48
- OP T ≤ 48



#4. Follow greedy coloring algorithm for the following graph. You can add more colors to the palette if needed.

Palette

Colors:	a	b	c	d
v1				
v2				
v3				
v4				
v5				
v6				
v7				
v8				
v9				
v10				



On the graph itself – indicate for each vertex its resulting color. On the palette – if vertex cannot be colored in a certain color, shade that square [same as in our slides]

-I couldn’t really draw another graph on here without messing up the one I have for reference so here’s how I’d color the graph:

We can make colors a, b, c.. and add colors as we need them. From the graph I would go like

v1=a

v2=b

v3=c

v4=a

v5=b

v6=c

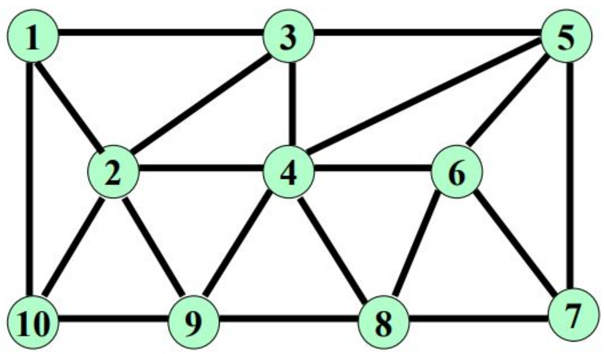
v7=a

v8=b

v9=c

v10=d

#5. Find Maximal Independent Set of this graph by LUBY's algorithm. Explain your steps.



We define a set, C = set of all vertices

C = {1,2,3,4,5,6,7,8,9,10}

L = {set of vertices in the maximal independent set}

Now, we will insert a vertex into the set L if all the vertices' neighbor's number is higher than that of it. So first we insert 1 into L, then we delete all the neighbor of 1 and also 1 from C.

C = {4,5,6,7,8,9}

L = {1}

Next step,

C = {7}

L = {1,4}

Last step,

C = {}

L = {1,4,7}

#6. Do branch-and-bound technique to generate all maximal independent sets (=leaves). Left child should be graph G1 = G – vertex. Right child: G2 = G – N(vertex). Do two levels – see next page