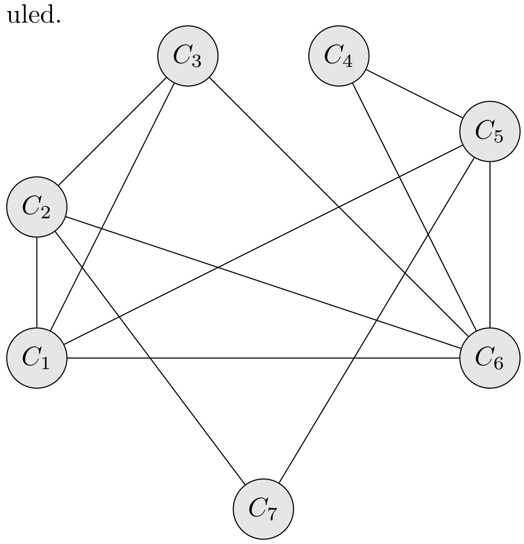
Scheduling Solution

hxxz46

A.3.1
Let the nodes of the graph correspond to the classes which need to be scheduled. An edge between two nodes indicates that there is some student who is in both classes. Therefore, these classes cannot be scheduled in the same time slot. The colour of a node will correspond to the time slot in which it will be scheduled.



If the graph can be coloured using four colours, all the classes can be scheduled during one day.

A.3.2

$$C_1, C_2, C_3, C_5, C_4, C_6, C_7$$

1. As no vertices have been visited, the algorithm begins at C_1 and colours it.

2. The available vertices are the ones which are adjacent to C_1. These are C_2, C_3, C_5, C_6. We select the smallest one which is C_2.

3. The available vertices are the ones which are adjacent to C_1 and C_2 which have not previously been visited. These are C_3, C_5, C_6, C_7. We select the smallest which is C_3.

4. The available vertices are the ones which are adjacent to C_1,C_2 and C_3 and have not previously been visited. These are C_5, C_6, C_7. We select the smallest among these which is C_5.

5. We now consider the neighbours of C_1, C_2, C_3 and C_5 which have not been previously coloured. These are C_4, C_6, C_7. We select C_4 as it is the smallest amongst these.

6. We now consider the neighbours of C_1, C_2, C_3, C_4 and C_5 which have not been previously coloured. These are C_6, C_7. We select C_6 as it is the smallest amongst these.

7. Finally, we consider the neighbours of C_1, C_2, C_3, C_4, C_5 and C_6 which have not been previously coloured. The only available vertex is C_7 which we select and colour.

A.3.3
$C_1 = 1$
$C_2 = 2$
$C_3 = 3$
$C_4 = 1$
$C_5 = 2$
$C_6 = 4$
$C_7 = 1$

1. First we visit C_1, no nodes have been coloured so we assign it 1, the smallest colour.

2. Next we visit C_2, it is adjacent to C_1 which has a colour of 1. Therefore, we assign it 2, the next smallest colour.

3. Next we visit C_3, it is adjacent to C_1 which has a colour of 1 and C_2 which has a colour of 2. Therefore, we assign it 3, the next smallest colour.

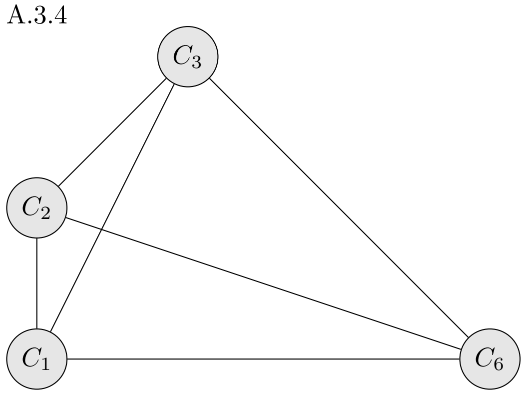
4. Next we visit C_5, it is adjacent to C_1 which has a colour of 1 and C_3 which has a colour of 3. Therefore, we assign it 2, the smallest available colour.

5. The algorithm colours C_4 next which is adjacent only to C_5 and nodes which have not yet been coloured. As C_5 has a colour of 2, we can assign C_4 1.

6. The algorithm selects C_6 which is adjacent to C_1, C_2, C_3, C_4 and C_5. There is a vertex in this set which has been coloured with 1, 2 and 3. Therefore, C_6 is coloured with 4.

7. C_7 is adjacent to C_2 and C_5. Both C_2 and C_5 have a colour of 2, therefore C_7 can be assigned a colour of 1.

A.3.4



The chromatic number of G is 4.

There is a connected subgraph on the above vertices. Therefore, it will require 4 colours to properly colour this subgraph as every node is adjacent to every other node. This means that the chromatic number of G,$\chi(G) \geq 4$. However, running the colouring algorithm on the graph produces an output that states the chromatic number of G,$\chi(G) \leq 4$. As we have $\chi(G) \leq 4$ and $\chi(G) \geq 4$, we can conclude that $\chi(G) = 4$.