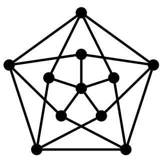
Universtiy of Queensland
School of Mathematics and Physics

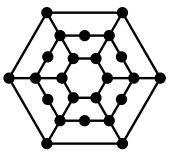
MATH2302 Discrete Mathematics II
Semester 2 2025
Problem Set 4

Michael Kasumagic, 44302669
Applied Class #1
Due 3pm Friday 28 October 2025

Question 1: 6 marks

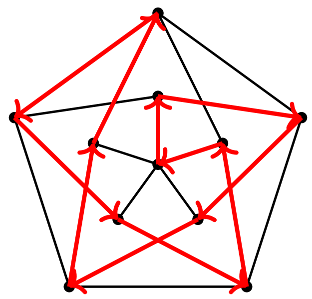
For each of the graphs shown below, determine whether it is Hamiltonian. Justify your answer.

(a) 

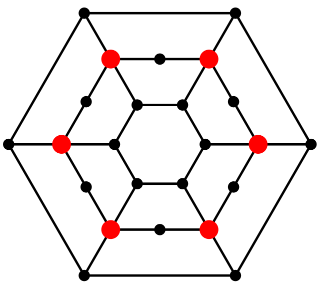
(b) 

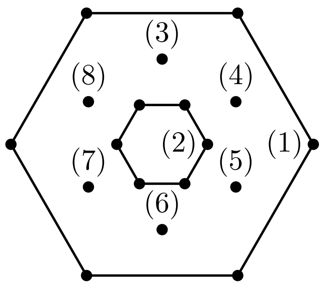
Solution:

Graph (a) is Hamiltonian. I present one such spanning cycle:



Graph (b) is not Hamiltonian. We show this by considering Theorem 26.85. Consider the set S, which contains only the highlighted vertices. Then consider the graph $G - S$, in particular, we count the number of components the graph has.





We highlighted 6 vertices for $S \subseteq G$, hence $|S| = 6$. The graph $G - S$, has componets: the outer ring, the inner ring, and the 6 isolated vertices. Hence, $|\text{Comp}(G - S)| = 8$.

$$|S| = 6 < 8 = |\text{Comp}(G - S)|\,.$$

Therefore, by Theorem 26.85, the graph is not Hamiltonian.

Question 2: 6 marks

For any integer $n \geq 4$, let K_n be the complete graph on vertices $1, 2, \ldots, n$. Let G_n be the graph obtained from K_n by deleting two edges $\{1, 2\}$ and $\{3, 4\}$. What is the vertex chromatic number of G_n? Justify your answer.

Solution:

Let K_n be the complete graph on vertices $\{1, 2, \ldots, n\}$, and let G_n be the graph obtained from K_n by deleting the two edges $\{1, 2\}$ and $\{3, 4\}$. We seek the vertex chromatic number, $\chi(G_n)$, for all integers $n \geq 4$.

Claim. $\chi(G_n) = n - 2$ for all $n \geq 4$.

Proof.
We'll start by finding an upper bound.

Define a colouring as follows.

$$c(1) = c(2) = A, \quad c(3) = c(4) = B,$$

where $A \neq B$. For the remaining vertices $i \in \{5, 6, \ldots, n\}$, assign a distinct, new colour,

$$c(i) = C_i$$

We'll check the colouring:

- Vertices 1 and 2 have the same colour, A. This is allowed because the edge $\{1, 2\}$ was deleted, so 1 and 2 are not adjacent in G_n.
- Vertices 3 and 4 have the same colour, B. This is allowed because the edge $\{3, 4\}$ was deleted, so 3 and 4 are not adjacent in G_n.
- Vertex 1 is adjacent to every vertex except 2, so no other vertex may use colour A.
- Vertex 2 is adjacent to every vertex except 1, so no other vertex may use colour A.
- Vertex 3 is adjacent to every vertex except 4, so no other vertex may use colour B.
- Vertex 4 is adjacent to every vertex except 3, so no other vertex may use colour B.
- For $i, j \in \{5, 6, \ldots, n\}$ with $i \neq j$, the edge $\{i, j\}$ is still present, so vertices i and j are adjacent. Therefore, they must be coloured differently. By construction we give each a distinct colour, C_i.
- Each vertex $i \in \{5, 6, \ldots, n\}$ is adjacent to $1, 2, 3, 4$, so it cannot reuse A or B. Which is why new colours C_i were required.

So, how many colours did we use?

$$\underbrace{A}_{\{1,2\}} + \underbrace{B}_{\{3,4\}} + \underbrace{(n-4) \text{ distinct colours } C_5, \ldots, C_n}_{\text{for vertices } 5,\ldots,n} = 2 + (n-4) = n - 2.$$

Therefore G_n can be coloured with, at most, $n - 2$ colours, so

$$\chi(G_n) \leq n - 2.$$

Next, we'll try find a lower bound.
Suppose, for contradiction, that there is a proper colouring of G_n using at most $n - 3$ colours.

We'll call this number of colours, k. So $k \leq n-3$.

Since there are n vertices and only $k \leq n-3$ colours, by the pigeonhole principle at least one colour is used on two or more distinct vertices. Any two vertices that share a colour must be non-adjacent. In G_n the only non-adjacent pairs are $\{1,2\}$ and $\{3,4\}$, where we removed edges from K_n. Every other pair of distinct vertices is still adjacent.

Therefore, in any colouring with at most $n-3$ colours, one of the following must happen:

$$1 \text{ and } 2 \text{ share a colour,} \quad \text{or} \quad 3 \text{ and } 4 \text{ share a colour.}$$

Without loss of generality, assume 1 and 2 share a colour, call it A.

Now, we'll consider the remaining $n-2$ vertices

$$\{3,4,\ldots,n\}.$$

We still have, at most, $k-1$ new colours available apart from A. Since A is already used on $\{1,2\}$ and cannot be reused on any other vertex (every other vertex is adjacent to 1 or 2). Since $k \leq n-3$, we have $k-1 \leq (n-3)-1 = n-4$.
So we are trying to colour $n-2$ vertices using at most $n-4$ colours.
Again, by pigeonhole, among these $n-2$ vertices there must be some colour that appears on at least two of them. Let those two vertices be x and y. As before, two vertices can only share a colour if they are non-adjacent. Among the set $\{3,4,\ldots,n\}$, the only non-adjacent pair is $\{3,4\}$.
Therefore 3 and 4 must also receive the same colour, say B.

Now look at the vertices $\{5,6,\ldots,n\}$. There are $(n-4)$ of them. In G_n they form a complete subgraph (every edge between them is still present), and each of these vertices is also adjacent to vertices $1,2,3,4$. This has two consequences:

- No two of $5,6,\ldots,n$ can share a colour with each other, because they are pairwise adjacent.
- None of $5,6,\ldots,n$ can reuse the colour A (shared by $1,2$) or the colour B (shared by $3,4$), because each of $5,6,\ldots,n$ is adjacent to all of $1,2,3,4$.

Hence, vertices $5,6,\ldots,n$ each require their own distinct new colour, different from A and different from B. That already forces $(n-4)$ new colours.
Together with colours A and B, the total number of colours required is at least

$$2+(n-4) = n-2.$$

This contradicts the assumption that we could colour G_n using at most $n-3$ colours.
Therefore no proper colouring of G_n can use fewer than $n-2$ colours, so

$$\chi(G_n) \geq n-2.$$

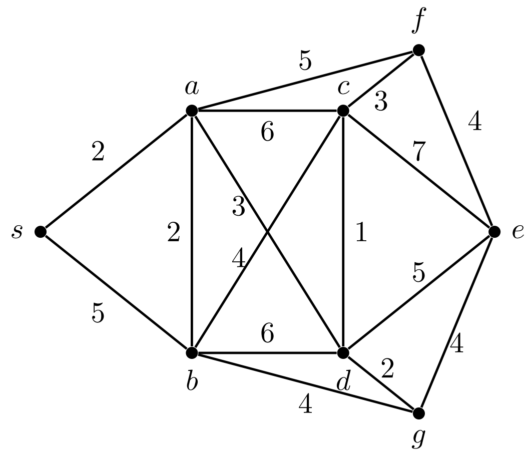
Therefore, we have $\chi(G_n) \leq n-2$ and $\chi(G_n) \geq n-2$. Hence,

$$\chi(G_n) = n-2 \quad \text{for all } n \geq 4.$$

$\square$

Question 3: 8 marks

Use Dijkstra's algorithm to determine the distance from s to each other vertex in the weighted graph shown below, and state a shortest path from s to e.



Solution:

s	a	b	c	d	f	g	e	Vertex Added to S
$\boxed{0}$	$(2,s)$	$(5,s)$	∞	∞	∞	∞	∞	s
	$\boxed{(2,s)}$	$(4,a)$	$(8,a)$	$(5,a)$	$(7,a)$	∞	∞	a
		$\boxed{(4,a)}$	$(8,a)$	$(5,a)$	$(7,a)$	$(8,b)$	∞	b
			$(6,d)$	$\boxed{(5,a)}$	$(7,a)$	$(7,d)$	$(10,d)$	d
			$\boxed{(6,d)}$		$(7,a)$	$(7,d)$	$(10,d)$	c
					$\boxed{(7,a)}$	$(7,d)$	$(10,d)$	f
						$\boxed{(7,d)}$	$(10,d)$	g
							$\boxed{(10,d)}$	e

From the final entry in the table, we can see that the shortest path from s to e has distance 10. Starting at the end, e, we can work our way back to d. Back to a. Then back to the starting point s. Therefore, the shortest path is

$$s \to a \to d \to e \quad \text{with} \quad d(s,e) = 10.$$

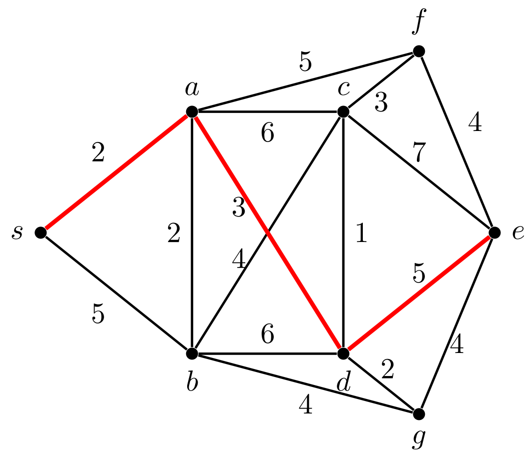


Figure 1: An illustraion of the shortest path, $s \to a \to d \to e$ which has distance $= 10$

Question 4: 10 marks

Use Algorithm 33.113 from the notes to find a maximum weight perfect matching in the weighted complete bipartite graph G with parts $V_1 = \{v_1, v_2, ..., v_6\}$ and $V_2 = \{u_1, u_2, ..., u_6\}$. The weight $w(v_iu_j)$ of the edge v_iu_j is given by $w(v_iu_j) = m_{ij}$, where $M = [m_{ij}]$ is given by

$$\begin{bmatrix} 7 & 5 & 9 & 4 & 7 & 6 \\ 6 & 9 & 9 & 7 & 5 & 5 \\ 5 & 6 & 4 & 8 & 9 & 7 \\ 8 & 7 & 5 & 6 & 8 & 4 \\ 6 & 5 & 7 & 5 & 9 & 8 \\ 7 & 6 & 6 & 9 & 5 & 9 \end{bmatrix}.$$

Solution:

We are given a complete weighted bipartite graph, G, with parts

$$V_1 = \{v_1, v_2, v_3, v_4, v_5, v_6\}, \quad V_2 = \{u_1, u_2, u_3, u_4, u_5, u_6\}.$$

The weight, $w(v_iu_j)$ is given by $v_iu_j \mapsto m_{ij}$, and m_ij are entries in the matrix

$$M = \begin{bmatrix} 7 & 5 & 9 & 4 & 7 & 6 \\ 6 & 9 & 9 & 7 & 5 & 5 \\ 5 & 6 & 4 & 8 & 9 & 7 \\ 8 & 7 & 5 & 6 & 8 & 4 \\ 6 & 5 & 7 & 5 & 9 & 8 \\ 7 & 6 & 6 & 9 & 5 & 9 \end{bmatrix}.$$

We're now going to apply Algorithm 33.113 to find a perfect matching $M^* \subseteq E$ of maximum total weight.

1) Initialise a feasible labelling.
By Algorithm 33.113 (1), we set

$$L(u_j) = 0 \quad \forall\, u_j \in V_2,$$

and for each $v_i \in V_1$,

$$L(v_i) = \max_{u_j \in V_2} w(v_iu_j).$$

So, in our case,

$$L(u_1) = 0, \quad L(u_2) = 0, \quad L(u_3) = 0, \quad L(u_4) = 0, \quad L(u_5) = 0, \quad L(u_6) = 0$$

and

$$L(v_1) = 9, \quad L(v_2) = 9, \quad L(v_3) = 9, \quad L(v_4) = 8, \quad L(v_5) = 9, \quad L(v_6) = 9.$$

2) Construct the subgraph H_L.
The subgraph H_L consists of all edges satisfying

$$L(v_i) + L(u_j) = w(v_iu_j).$$

Because $L(u_j) = 0$, $\forall u \in V_2$, these are exactly the edges where $w(v_iu_j) = L(v_i)$, i.e., the maximum-weight edges in each row of M. Therefore,

$$E(H_L) = \{v_1u_3, v_2u_2, v_2u_3, v_3u_5, v_4u_1, v_4u_5, v_5u_5, v_6u_4, v_6u_6\}.$$

Remark: I find this by reading from the matrix. Take v_4, which corresponds to the 4th row of M. We know that $L(v_4) = 8$, and there are two entries which have this weighting, the entry corresponding to u_1 (the first column, i.e. m_{41}) and the entry corresponding to u_5 (the fifth column, i.e. (m_{44})).

3) Find a maximum matching M in H_L using Algorithm 33.109.
Algorithm 33.109 (the augmenting path algorithm) searches for augmenting paths in H_L to build a maximum matching. A greedy attempt gives:

$$M = \{v_4u_1,\ v_6u_4,\ v_1u_3,\ v_2u_2,\ v_3u_5\}.$$

Here, v_5 remains unmatched because both v_3 and v_5 connect only to u_5. Hence, M is not perfect.

4) Build an alternating tree and adjust labels.
Since M is not perfect, choose an unmatched vertex $v_5 \in V_1$. Construct the alternating tree T rooted at v_5 in H_L. When the tree cannot expand further, compute

$$m_L = \min\{L(v) + L(u) - w(vu) : v \in V_1 \cap V(T),\ u \in V_2 \setminus V(T)\}.$$

Subtract m_L from each $L(v)$ for $v \in V_1 \cap V(T)$ and add m_L to each $L(u)$ for $u \in V_2 \cap V(T)$. This preserves feasibility and introduces at least one new tight edge connecting T to $V_2 \setminus V(T)$.

Step 5. Repeat Steps 2-4 until H_L admits a perfect matching.
After performing the label adjustments and re-running Algorithm 33.109, we eventually obtain a perfect matching M^* in H_L. By Theorem 33.112, this matching is a maximum weight perfect matching in G.

The algorithm yields the perfect matching

$$M^* = \{v_1u_3,\ v_2u_2,\ v_3u_4,\ v_4u_1,\ v_5u_5,\ v_6u_6\}.$$

The total weight is

$$w(M^*) = 9 + 9 + 8 + 8 + 9 + 9 = 52.$$

Using Algorithm 33.113, we find that the maximum weight perfect matching for the given bipartite graph is

$$M^* = \{v_1u_3,\ v_2u_2,\ v_3u_4,\ v_4u_1,\ v_5u_5,\ v_6u_6\},$$

with total weight 52. Therefore, M^* is the optimal assignment.