1. Let G and H be two undirected graphs. A mapping $f : V(G) \to V(H)$ is a *graph homomorphism* if for all $u, v \in V(G)$ the following holds: if uv is an edge in G, then $f(u)f(v)$ is an edge in H. For a fixed graph H (that is, H is not part of the input), the problem H-COLOURING takes as input a graph G and is to decide if there exists a homomorphism from G to H.

   (a) The H-COLOURING problem generalizes the k-COLOURING problem. Explain why this is the case.

   Solution: Let $H = K_k$ (the complete graph with k vertices). Then, if there is an homomorphism between G and H, it is necessary that G is k-colourable. Why? Suppose that $f : V(G) \to V(H)$ is a graph homomorphism and assign distinct colours to each vertex of H, which clearly defines a k-colouring. We now colour G: colour vertex $v \in V(G)$ with the colour of $f(v) \in V(H)$. This is a valid k-colouring. We check this: suppose $xy \in E(G)$. Then $f(x)f(y) \in E(H)$. Thus x and y must have distinct colours.

   (b) Let H be a graph that has a vertex u with a self-loop (so $uu \in E(H)$). Give a constant-time algorithm for H-COLOURING.

   Solution: Let G be the input graph and define $f : V(G) \to V(H)$ by $x \mapsto u$. We claim that f is a graph homomorphism. Indeed, let $x, y \in V(G)$. Then $f(x)f(y) = uu \in E(H)$. Thus the output of our algorithm is *yes*.

   (c) Let H be a bipartite graph (so H has no self-loops). Give a polynomial-time algorithm for H-COLOURING.

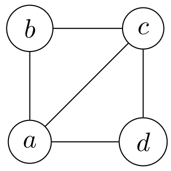
   Solution: We first claim that no graph homomorphism exists between a non-bipartite graph and a bipartite graph. Indeed, let $f : A \to B$ be such a homomorphism. A graph is bipartite if and only if it is 2-colourable. Let $g : B \to K_2$ be a graph homomorphism (which exists as B is 2-colourable, mapping the vertices to its colour). But then $g \circ f : A \to K_2$ is also a graph homomorphism (a composition of graph homomorphisms is a graph homomorphism). Thus A is bipartite; a contradiction.

   Thus, if G is non-bipartite (which we can greedily check in $O(n^2)$ time), then we output *no*.

   Now we consider G bipartite. Let $G_1, G_2 \subset V(G)$ be the partitions of G. If H contains no edges, then there is a graph homomorphism if and only if G contains no edges. Now let $uv \in E(H)$. We can now construct the homomorphism $f : V(G) \to V(H)$ such that $f(G_1) = u$ and $f(G_2) = v$. Thus if H has edges, the answer is *yes*.

   (d) Let H be the diamond, which is the graph obtained from the complete graph on four vertices after removing an edge. What is the computation complexity of H-COLOURING?

   Solution: We draw the diamond graph below.



   The diamond graph is 3-colourable. Indeed, define $f : V(H) \to V(K_3) = \{1, 2, 3\}$ by $f(b) = f(d) = 1$, $f(a) = 2$, and $f(c) = 3$. Thus if G is H-colourable, it is 3-colourable. That is, 3-COLOURING *reduces* to H-COLOURING. But 3-COLOURING $\in$ NPC, so H-COLOURING $\in$ NPC. We have taken for granted that H-COLOURING is polynomial-time verifiable, but we can just greedily check that a homomorphism is valid.

2. (a) Let $\mathcal{G}$ be the class of trees. Is $\mathcal{G}$ hereditary? If so, determine $\mathcal{F}_{\mathcal{G}}$.

Solution: $\mathcal{G}$ is not hereditary. Consider P_3. Deleting the middle vertex (with degree 2) leaves $2P_1$, which is a tree.

(b) Let $\mathcal{G}$ be the class of forests. Is $\mathcal{G}$ hereditary? If so, determine $\mathcal{F}_{\mathcal{G}}$.

Solution: $\mathcal{G}$ is indeed hereditary. We have $\mathcal{F}_{\mathcal{G}} = \{C_3, C_5, C_7, \ldots\}$.

(c) Let $\mathcal{G}$ be the class of paths. Is $\mathcal{G}$ hereditary? If so, determine $\mathcal{F}_{\mathcal{G}}$.

Solution: No, see the example in (a).

(d) Let $\mathcal{G}$ be the class of linear forests. Is $\mathcal{G}$ hereditary? If so, determine $\mathcal{F}_{\mathcal{G}}$.

Solution: $\mathcal{G}$ is indeed hereditary, and $\mathcal{G}_{\mathcal{F}} = \{C_3, C_5, C_7, \ldots\} \cup \{K_{1,3}\}$.

(e) Let $\mathcal{G}$ be the class of complete graphs. Is $\mathcal{G}$ hereditary? If so, determine $\mathcal{F}_{\mathcal{G}}$.

Solution: $\mathcal{G}$ is hereditary, and $\mathcal{F}_{\mathcal{G}} = \{2P_1\}$.

(f) Let $\mathcal{G}$ be the class of graphs that are disjoint union of complete graphs. Is $\mathcal{G}$ hereditary? If so, determine $\mathcal{F}_{\mathcal{G}}$.

Solution: $\mathcal{G}$ is hereditary with $\mathcal{F}_{\mathcal{G}} = \{P_3\}$.

(g) A graph is *chordal* if every subgraph C of G that is a cycle on at least four vertices has a *chord*, that is, an edge between two non-adjacent vertices of C. Is $\mathcal{G}$ hereditary? If so, determine $\mathcal{F}_{\mathcal{G}}$.

Solution: $\mathcal{G}$ is hereditary with $\mathcal{F}_{\mathcal{G}} = \{C_4, C_5, C_6, \ldots\}$.

3. Let $\mathcal{G}$ be a hereditary graph class for which $\mathcal{F}_{\mathcal{G}}$ is finite. Describe a polynomial-time algorithm for deciding if a given graph belongs to $\mathcal{G}$.

Solution: A greedy algorithm works here: for each $H \in \mathcal{F}_{\mathcal{G}}$ check each $G[A]$ such that $A \subset V(G)$ and $|A| = |H|$. An upper bound on the running time is

$$\sum_{H \in \mathcal{F}_{\mathcal{G}}} \binom{|G|}{|H|} = O(n^k)$$

where $k = \max\{|H| : H \in \mathcal{F}_{\mathcal{G}}\}$.

4. (a) Is the class of C_3-free graphs closed under edge deletion?

Solution: Yes.

(b) Is the class of K_{10}-free graphs closed under edge deletion?

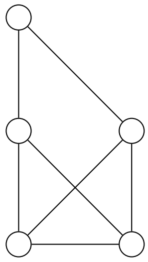
Solution: Yes.

(c) Are the classes of K_4-free graphs and K_4-subgraph-free graphs the same?

Solution: Yes.

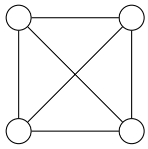
(d) Are the classes of K_4-free graphs and K_4-contraction-free graphs the same?

Solution: No. Consider the following graph.



(e) Is the class of diamond-free graphs closed under edge deletion?

Solution: No. Consider the following graph.



(f) Is the class of P_6-free graphs closed under edge contraction?

Solution: `todo`

(g) Is the class of P_6-subgraph-free graphs closed under edge contraction?

Solution: `todo`

5. Let H be a graph. Prove that the class of H-free graphs is closed under edge deletion if and only if H is a complete graph.

6. For $i \geq 0$, the graph H_i is obtained after subdividing one specific edge of the claw i times, so $H_0 = K_{1,3}$, whereas H_1 is also known as the chair. Is the class $\mathcal{G} = \{H_i : i \geq 0\}$ hereditary? If so, determine its set of minimal forbidden induced subgraphs $\mathcal{F}_{\mathcal{G}}$.

7. Let C_5 denote the 5-vertex cycle. Determine if the class of C_5-free graphs is closed under edge contractions.

8. Let C_3 denote the 3-vertex cycle. Robertson and Seymour gave a cubic-time algorithm for deciding if a graph contains some fixed graph H as a minor. Give a polynomial-time algorithm for deciding if a given graph contains C_3 as a minor that does not rely on this result.

9. Determine the clique-width of the graph $G = 2P_5$, that is, G is the disjoint union of two 5-vertex paths.

10. Let G be a graph and S be a subset of $V(G)$. Let $G-S$ be the graph obtained from G after removing the vertices of S. Recall that S is a feedback vertex set if $G-S$ is a forest. The Feedback Vertex Set problem is to determine the size of a smallest feedback vertex set of a given graph. Give a polynomial-time algorithm for Feedback Vertex Set for $10P_1$-free graphs.

11. A graph is co-bipartite graph if it is the complement of a bipartite graph, or in other words, if its vertex set can be partitioned into two (possibly empty) sets A and B, such that A and B are cliques. Prove that co-bipartite graphs have unbounded clique-width.

12. Let P be the path on vertices $u_1, \ldots, u_5$ in that order. We obtain a graph G from P by adding four new vertices s_1, s_2, t_1, t_2 with edges $u_1s_1, u_1s_2, u_5t_1, u_5t_2$. Determine the treewidth and the pathwidth of G.

13. Let G be a K_3-free chordal graph. Give a constant-time algorithm for deciding if G has a 5-colouring.

14. A graph is a split graph if its vertex set can be partitioned into two (possible empty) sets C and I such that C is a clique and I is an independent set. Express the property of being a split graph in MSO_2.

15. As the property of being split can be expressed in MSO_1, we can decide in polynomial time if a graph from a class of bounded clique-width is a split graph (by applying the meta-theorem on slide 5 of Lecture 19). However, we can recognize *any* split graph in polynomial time. Prove this.