University Côte d'Azur — Graphs
Master 2 : Informatique Fondamentale — November, 2025

Final exam (part 1) : 1 hour

No electronic/manuscript document is allowed. You can answer in french or english. Each answer must be explained.

1 Glossary

Let $G = (V, E)$ be a graph.

— A **vertex cover** of G is a set Q of vertices "touching" all edges, i.e., a set $Q \subseteq V$ such that $e \cap Q \neq \emptyset$ for all $e \in E$.

— A **matching** of G is a set M of pairwise disjoint edges, i.e., a set $M \subseteq E$ such that $e \cap f = \emptyset$ for all $e \neq f \in E$.

— A matching $M \subseteq E$ is **maximal** if, for every $e \in E \setminus M$, $\{e\} \cup M$ is not a matching.

— An **independent set** of G is a set S of vertices that are pairwise non-adjacent, i.e., a set $S \subseteq V$ such that, for all $u, v \in S$, $\{u, v\} \notin E$.

— A vertex is **isolated** if its degree is 0, i.e., it has no neighbour.

2 Basics

Question 1 *Give an application (in the "real life") of the problem of finding a minimum vertex cover of a graph.*

Question 2 *Prove that if any graph G admits a matching M of size ℓ, then every vertex cover of G has size at least ℓ.*

Question 3 *Let S_n be the graph obtained from n pairwise disjoint edges $(e_i = \{u_i, v_i\})_{1 \leq i \leq n}$ by adding a new vertex c adjacent to each vertex u_i, $1 \leq i \leq n$.*

— *Draw S_n ;*

— *Give a vertex cover of S_n of size n ;*

— *Prove that this is a vertex cover of minimum size of S_n (use Question 2).*

Question 4 *Describe (in at most 2-3 sentences) a polynomial-time algorithm that computes a maximal matching in a graph $G = (V, E)$. Give its time complexity as a function of $|E|$.*

Question 5 *Describe (in at most 2 sentences) a brute force algorithm that decides whether a graph $G = (V, E)$ admits a vertex cover of size $k \in \mathbb{N}$. What is its time complexity ?*

Question 6 *Describe (in a few sentences) a branching algorithm that decides whether a graph $G = (V, E)$ admits a vertex cover of size $k \in \mathbb{N}$ in time $O(2^k poly(n))$.*

Question 7 *Prove that, if $Q \subseteq V$ is a vertex cover of a graph $G = (V, E)$, then its complement $V \setminus Q$ is an independent set.*

3 Crown decomposition

Question 8 *Describe the general principle and time-complexity of a kernelization algorithm.*

Definition 1 *A* **crown decomposition** *of a graph $G = (V, E)$ is a partition of V into three parts C (the crown), H (the head) and R (the remainder), such that*

1. *C and H are non-empty.*
2. *C is an independent set.*
3. *There are no edges between vertices of C and R. That is, H separates C and R.*
4. *Let E_0 be the set of edges between vertices of C and H. Then E_0 contains a matching M of size $|H|$. In other words, G contains a matching of H into C.*

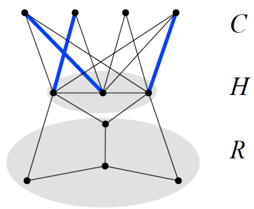

FIGURE 1 – Illustration of a crown decomposition with M in bold blue.

Question 9 *Let G be a graph with a crown decomposition (C, H, R) and let Q be a vertex cover of G. Show that Q contains at least $|H|$ vertices of $H \cup C$, i.e. $|(C \cup H) \cap Q| \geq |H|$.*

Question 10 *Let G be a graph with a crown decomposition (C, H, R). Show that G has a vertex of size at most $k \in \mathbb{N}$ if and only if $G[R]$ has a vertex cover of size at most $k - |H|$.*

For the last question, we admit the following theorem.

Theorem 1 *Let G be a graph with n vertices and without isolated vertices and with at least $3k + 1$ vertices. There is a polynomial time (say $O(n^3)$) algorithm that either*

— *finds a matching of size at least $k + 1$ in G ; or*

— *finds a crown decomposition of G.*

Question 11 — *What is the goal of Algorithm 1 ?*

— *Describe how this kernelization algorithm works.*

— *What is its time complexity (as a function of the size n of G and of k) ?*

Algorithm 1 Algo(G, k)

Require: A graph $G = (V, E)$ and $k \in \mathbb{N}$.
Ensure: ? ? ?

Remove isolated vertices of G.
if G has at least $3k + 1$ vertices **then**
  Apply the algorithm of Theorem 1 to G.
  if G admits a matching of size at least $k + 1$ **then**
    return False
  else
    Let (C, H, R) be a crown decomposition of G.
    return Algo$(G[R], k - |H|)$
else
  Use the branching algorithm of Question 6 on (G, k).