Examen du cours 2-38-2 du MPRI 2020
Algorithmique et combinatoire des graphes géométriques

The duration of this exam is 3 hours. You have a 15 minutes buffer at the end of the exam, so until 4pm, to scan it and send it to both of us: arnaud.demesmay@u-pem.fr and vincent.pilaud@lix.polytechnique.fr.

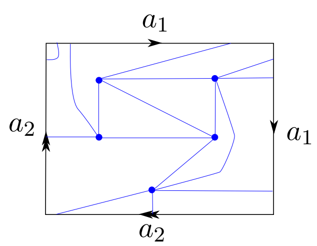
This is an open-book exam: you are free to consult the lecture notes or any source online. You are not allowed to communicate with each other.

The language can be either English or French.

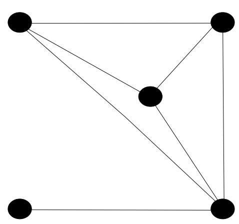
The exercises are independent and can be treated in any order.

1 Graphs on surfaces (ADM)

Exercice 1. Recall that a cellular embedding is an embedding where all the faces are disks, and that a non-orientable surface of genus g is a surface with polygonal scheme $a_1a_1a_2a_2\dots a_ga_g$. A convenient way to represent a graph on a non-orientable surface is to draw it on top of this polygonal scheme. For example, here is a cellular embedding of K_5 on a non-orientable surface of genus two.



1. Provide an explicit cellular embedding of the graph pictured below on a non-orientable surface of genus 3.



2. Let G be a simple graph with v vertices, e edges cellularly embedded on a non-orientable surface of genus g. Prove that $g \leq e - v + 1$.

3. Let G be a simple graph with v vertices and e edges, and let g_1 be the smallest genus of a non-orientable surface on which G embeds. Prove that for any g such that $g_1 \leq g \leq e - v + 1$, G can be cellularly embedded on a non-orientable surface of genus g.

4. In particular, G can always be cellularly embedded on a non-orientable surface of genus $e - v + 1$. Provide a linear-time algorithm to compute such an embedding.

Exercice 2. In this exercise, we look at a graph G cellularly embedded on an orientable surface of genus g, where each edge uv has an orientation from its **start** to its **end**: either $u \to v$ or $v \to u$[1]. This induces an orientation on the edges of the dual graph, as pictured in Figure 1. We will only be considering oriented cycles in this exercise: a cycle in G is a sequence of oriented edges $(v_1 \to v_2), (v_2 \to v_3) \dots (v_k \to v_1)$ with no repeated vertices. Recall that a cycle is **contractible** if it can be continuously deformed to a point on the surface, or equivalently if it bounds a closed disk.

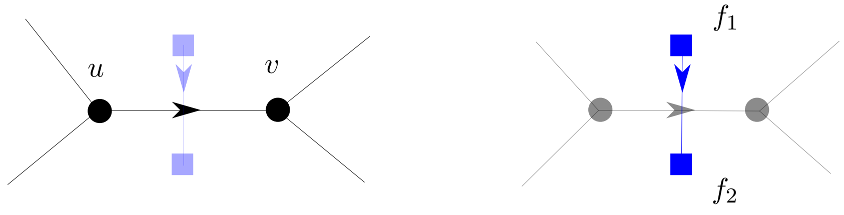


Figure 1: The start of the dual edge (here f_1) lies to the left of the primal edge.

1. List all the cycles of the toroidal graph pictured below, as well as all the cycles of its dual. To make this easier, edges are named. Which of these cycles are contractible?

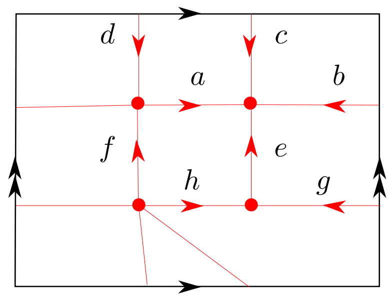


Figure 2: An oriented graph (in red) embedded on a torus. To be clear: the black edges are not part of the graph and are just here to represent the torus (top identified to bottom and left identified to right).

2. Let e be an edge so that the dual edge e^* belongs to a cycle. Prove that e can not belong to a contractible cycle.

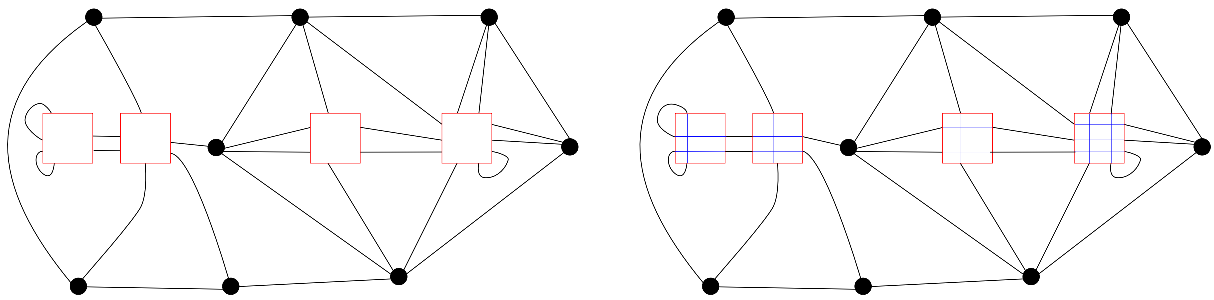
3. Suppose that the dual graph G^* has no cycle. Prove that if G has a contractible cycle, G^* has a **sink**, i.e., a vertex v where all the incident edges are oriented towards v, or a **source**, i.e., a vertex v where all the incident edges are oriented away from v.

4. Still supposing that the dual graph G^* has no cycle, deduce that if G has a contractible cycle, then it has a face whose boundary is a cycle.

5. Provide a linear-time algorithm to test whether an oriented graph G embedded on an orientable surface contains a contractible cycle.

[1]Orientations of loops are a bit annoying to define. To keep things simple, for the purpose of this exercise, we use the same definition: a loop based at u is oriented $u \to u$. In particular it has a unique possible orientation.

Exercice 3. We consider the following way of representing non-planar graphs with boxes. There are k disjoint squares called **boxes** drawn in the plane, and each side acts as a teleporter to the same point on the opposite side. A graph is embedded in the plane with k boxes if it is drawn without crossings in the plane when the edges are allowed to use these teleporters: when an edge intersects a point on the box, it continues on the same point on the opposite side. Note that each edge is allowed to use the same box any number of times. For example, here is a picture of a graph embedded in the plane with four boxes (left picture). Equivalently, a box is a way to hide a grid of crossings (see the right picture).



1. Provide an embedding of K_5 in the plane with a single box.
2. Prove that a graph can be embedded in the plane with g boxes if and only if it can be embedded on a surface of genus g.
3. Let G be a graph embedded on a surface of genus g. By the previous question, G can be embedded in the plane with g boxes. Find a function $f(g)$ so that the following strengthening holds (and prove it): G can be embedded in the plane with g boxes so that each edge of G crosses at most $f(g)$ boxes (counted with multiplicity). Any function (even non-polynomial) will do, but the smaller ones are worth more points.

2 Polytopes and geometry of binary trees (VP)

Exercice 4. For a polytope P, let $p_k(P)$ be its number of k-gonal 2-faces for each $k \geq 3$.

(1) Show that for a simple 3-polytope P, we have $\sum_{k\geq 3}(6-k)\cdot p_k(P) = 12$.

(2) Show that every simple 3-polytope contains at least four 2-faces each of which has at most five edges.

(3) Let $\mathcal{C} \subset \mathbb{R}^3$ be the set of points $(p_3(P), p_4(P), p_5(P))$ for all simple 3-polytopes P. Show that $\mathcal{C}$ is contained in the intersection X of the positive orthant $\mathbb{R}^3_{\geq 0}$ with the half-space defined by $3\,p_3 + 2\,p_4 + p_5 \geq 12$. Compute the vertices of the polyhedron X and provide simple polytopes P such that $(p_3(P), p_4(P), p_5(P))$ correspond to these vertices.

(4) Provide simple polytopes P such that $(p_3(P), p_4(P), p_5(P))$ correspond to other integer points in the hyperplane $3\,p_3 + 2\,p_4 + p_5 = 12$ (no need to find a polytope for each integer point, but give as much as you can).

Exercice 5. A *binary tree* is either the empty tree, or a node (called root) with two children (called left and right) which are themselves binary trees. We denote by $\mathcal{B}_n$ the set of binary trees with n nodes. Recall that the *inorder* (resp. *postorder*) on a binary tree T is the order obtained by labeling the root of T between (resp. after) labeling the left and right subtrees of T in inorder (resp. postorder). See Figure 3.

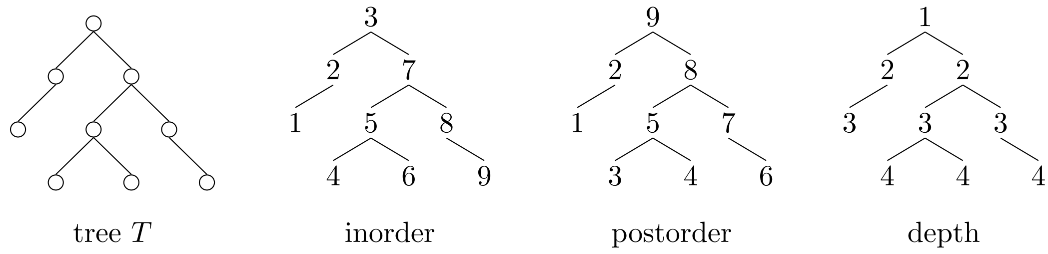


Figure 3: A tree, its inorder and postorder labelings, and its depths.

We call *depth* of a node n the number of nodes along the path from n to the root r (counting r and n, so that the root has depth 1). See Figure 3. The *depth vector* $d_{\text{in}}(T)$ (resp. $d_{\text{post}}(T)$) is the vector whose jth coordinate is the depth of the jth node of T in inorder (resp. postorder). For example, for the binary tree T of Figure 3, we have $d_{\text{in}}(T) = (3, 2, 1, 4, 3, 4, 2, 3, 4)$ and $d_{\text{post}}(T) = (3, 2, 4, 4, 3, 4, 3, 2, 1)$. We denote by $D_{\text{in}}(n)$ (resp. $D_{\text{post}}(n)$) the convex hull of the depth vectors $d_{\text{in}}(T)$ (resp. $d_{\text{post}}(T)$) for all binary trees $T \in \mathcal{B}_n$ with n nodes. The objective of this problem is to study the number of vertices of the polytopes $D_{\text{in}}(n)$ and $D_{\text{post}}(n)$.

1. List all binary trees T with 3 nodes and the corresponding depth vectors $d_{\text{in}}(T)$ and $d_{\text{post}}(T)$. What are the polytopes $D_{\text{in}}(3)$ and $D_{\text{post}}(3)$.

2. Recall why the numbers $b_n = |\mathcal{B}_n|$ satisfy $b_{n+1} = \sum_{i=0}^{n} b_i \cdot b_{n-i}$ and that the corresponding generating function $\mathrm{B}(x) = \sum_{n \geq 0} b_n x^n$ is given by $\mathrm{B}(x) = \frac{1-\sqrt{1-4x}}{2x}$. Compute b_n for $n \leq 6$.

3. Show that $D_{\text{in}}(n)$ has b_n vertices.

A *unary-binary tree* is a rooted tree where each node has either zero, one or two children. In other words, from a binary tree T, we obtain a unary-binary tree $\bar{T}$ by forgetting for each node with no sibling whether it was a left or a right child. We denote by $\mathcal{UB}_n$ the set of unary-binary trees with n nodes. We define the *postorder*, the *depth* and the *depth vector* $d_{\text{post}}(U)$ on a unary-binary tree U as for binary trees. See Figure 4, where $d_{\text{post}}(T_1) = d_{\text{post}}(T_2) = d_{\text{post}}(U) = (3, 2, 4, 3, 4, 3, 2, 1)$.

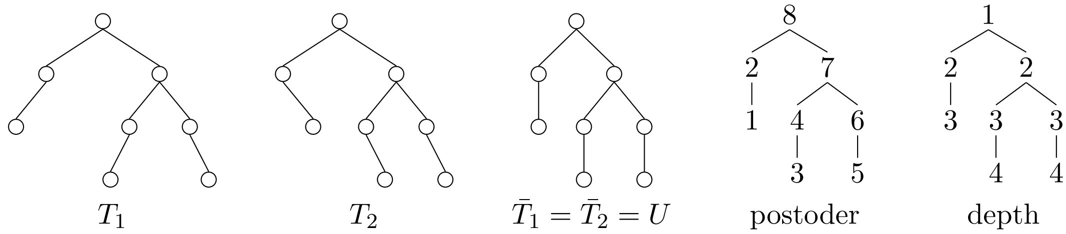


Figure 4: Two binary trees T_1 and T_2 sent on the same unary-binary tree U, the postorder on U, and the depth on U.

4. List all unary-binary trees T with at most 5 nodes and the corresponding depth vectors $d_{\text{post}}(T)$.

5. Give a recurrence formula for the numbers $ub_n = |\mathcal{UB}_n|$ and a closed expression for the corresponding generating function $\mathrm{UB}(x) = \sum_{n \geq 0} ub_n x^n$ similar to that of Question 2. Compute ub_n for $n \leq 6$.

6. Show that the cardinality of the set $\{d_{\text{post}}(T) \mid T \in \mathcal{B}_n\}$ is ub_n.

We say that a unary-binary tree is *monotone* when any node with no sibling has zero or one child. For instance, the unary-binary tree U of Figure 4 is monotone. We denote by $\mathcal{MUB}_n$ the set of monotone unary-binary trees with n nodes.

7. List all monotone unary-binary trees T with at most 5 nodes and the corresponding depth vectors $d_{\text{post}}(T)$.
8. Give a recurrence formula for the numbers $mub_n = |\mathcal{MUB}_n|$ and a closed expression for the corresponding generating function $\text{MUB}(x) = \sum_{n\geq 0} mub_n x^n$ similar to that of Questions 2 and 5. Compute mub_n for $n \leq 6$.
9. Show that $D_{\text{post}}(n)$ has mub_n vertices.

Exercice 6. The *integer point transform* of a subset $\mathbf{A}$ of $\mathbb{R}^n$ is the abstract multivariate generating series

$$\text{IPT}(A) = \sum_{a \in A \cap \mathbb{Z}^n} x^a,$$

where $x^a = x_1^{a_1} \cdots x_n^{a_n}$.

(1) Show that IPT is a valuation.

(2) Recall that a vector is *primitive* if its coordinates are relatively prime numbers. Show that the integer point transform of a polyhedral cone C generated by n linearly independent primitive integer vectors $v_1, \dots, v_n$ is given by

$$\text{IPT}(C) = \frac{1}{\prod_{i\in[n]}(1 - x^{v_i})}.$$

For a poset $\prec$ on $[n]$, we define the polyhedral cone

$$C(\prec) = \left\{ x \in \mathbb{R}^n_{\geq 0} \;\middle|\; \begin{array}{l} x_i \leq x_j \text{ for all } i \prec j \text{ with } i < j \\ x_i < x_j \text{ for all } i \prec j \text{ with } i > j \end{array} \right\}.$$

As usual, we see

- a permutation π of $[n]$ as the total order $\prec_\pi$ defined by $\pi_1 \prec_\pi \pi_2 \prec_\pi \cdots \prec_\pi \pi_n$
- a binary tree T with n nodes as the poset $\prec_T$ where $i \prec_T j$ if and only if i is a descendant of j in T when labeled in infix labeling (label first the left child in infix labeling, then the root, and then the right child in infix labeling).

(3) Compute $\text{IPT}(\prec_\pi)$ for all permutations π of $[3]$ and $\text{IPT}(\prec_T)$ for all binary trees T with 3 nodes.

(4) Show that the integer point transform of the cone $C(\prec_\pi)$ given by a permutation π of $[n]$,

$$\text{IPT}(C(\prec_\pi)) = \frac{\displaystyle\prod_{\substack{i\in[n-1]\\ \pi_i > \pi_{i+1}}} x_{\pi_i} \cdots x_{\pi_n}}{\displaystyle\prod_{i\in[n]} (1 - x_{\pi_i} \cdots x_{\pi_n})}$$

(5) Give and prove a similar formula for the integer point transform of the cone $C(\prec_T)$ given by a binary tree T with n nodes.

(6) Connect the integer point transforms of $C(\prec_T)$ and $C(\prec_\pi)$ for certain permutations π.