

Graphes, rayons et coloration

Un *graphe* $G = (V, E)$ est non orienté. Un *rayon* de G est une suite infinie de sommets $s_0, s_1, s_2, \dots$ telle que pour tout $i \geq 0$, la suite $s_0, \dots, s_i$ est un chemin simple. On dit que G vérifie la propriété (K) s'il est connexe, infini et localement fini (c'est-à-dire que pour tout sommet s , l'ensemble $N_s := \{x \mid \{s, x\} \in E\}$ est fini).

1. Montrer que si un graphe G admet un rayon, alors il est infini.
2. Un graphe $G = (V, E)$ est dit *localement borné* s'il existe une constante M telle que pour tout $s \in V$, $|N_s| < M$. Donner :
 - un exemple de graphe infini, connexe et localement borné ;
 - un exemple de graphe infini, connexe, localement fini mais pas localement borné.
3. Est-ce que chacune des trois propriétés de (K) sont nécessaires pour que le théorème

Tout graphe dénombrable qui vérifie (K) admet un rayon

soit vrai ?

4. Montrer qu'il existe un graphe G infini et localement fini tel que pour tout k , G possède un chemin simple de taille k mais que G n'admet pas de rayon.
5. Étant donné $G = (V, E)$ vérifiant (K) et $v \in V$, on pose $G_v = (V', E')$ avec $V' = V \setminus \{v\}$ et $E' = E \cap \{\{x, y\} \mid x \in V'\}$. Montrer que G_v est une union disjointe finie de graphes connexes dont au moins l'un vérifie (K).
6. Montrer que si G satisfait (K) alors il admet un rayon.

Application à la coloration. Soit $G = (\mathbb{N}, E)$ un graphe. Une k -coloration de G est une fonction $f : \mathbb{N} \rightarrow \{0, \dots, k-1\}$ telle que pour tout $\{x, y\} \in E$, $f(x) \neq f(y)$. On dit que G est k -coloriable s'il en existe une.

7. Montrer qu'un graphe dénombrable G est k -coloriable si et seulement si tous ses sous-graphes finis sont k -coloriables.

Solution : Graphes, rayons et coloration

1. Supposons que G admette un rayon $s_0, s_1, s_2, \dots$. Par définition, pour tout $i \geq 0$, la suite $s_0, \dots, s_i$ est un chemin simple, donc tous les sommets s_i sont distincts. L'ensemble $\{s_i \mid i \geq 0\}$ est infini, donc V est infini.
2. — **Localement borné :** la grille $\mathbb{Z}^2$ (chaque sommet a exactement 4 voisins) est infinie, connexe et localement bornée (avec $M = 5$).
 — **Localement fini mais pas localement borné :** l'arbre où le sommet n a $n + 1$ voisins : chaque sommet a un nombre fini de voisins, mais ce nombre est non borné.
3. On montre que chacune des trois propriétés est nécessaire en exhibant un contre-exemple pour chacune.
 - **Sans connexité :** une infinité de sommets isolés est infinie, localement finie, mais n'admet pas de rayon.
 - **Sans infinité :** un graphe fini connexe localement fini n'admet pas de rayon (un rayon requiert une infinité de sommets distincts).
 - **Sans finitude locale :** l'étoile $K_{1,\infty}$ (un centre relié à dénombrablement many feuilles) est infinie et connexe, mais tout chemin simple depuis le centre a longueur 1, donc il n'y a pas de rayon.
4. On construit un graphe en prenant pour $V = \mathbb{N}$ et en reliant le sommet 0 à tous les autres, et aucune autre arête. Ce graphe est localement fini (chaque feuille a un seul voisin ; le centre 0 a des voisins $\{1, 2, 3, \dots\}$ infinis, donc ce n'est pas localement fini).
 Pour un vrai contre-exemple : considérer les graphes à chenilles C_k (un chemin de longueur k avec des feuilles pendantes), et prendre leur union disjointe. Ce graphe est localement fini, possède des chemins simples arbitrairement longs, mais est non connexe.
 Pour un exemple connexe : construire un arbre infini en éventail, où depuis la racine partent des branches de longueur 1, 2, 3, ... reliées les unes aux autres. Toute suite infinie devrait rester dans une branche finie, ce qui est impossible, donc il n'y a pas de rayon.
5. Soit G satisfaisant (K) et $v \in V$. Comme G est localement fini, N_v est fini, donc supprimer v ne déconnecte G qu'entre les composantes qui étaient reliées uniquement via v . Le nombre de composantes de G_v est au plus $|N_v| + 1$, qui est fini.
 Chaque composante C_i est connexe par définition. Au moins l'une d'elles est infinie : en effet G est infini, $V' = V \setminus \{v\}$ est infini, et il n'y a qu'un nombre fini de composantes, donc l'une au moins est infinie. Cette composante infinie est connexe et localement finie (propriétés héritées de G), donc elle satisfait (K).
6. On raisonne par récurrence (construction d'un rayon par König) : on construit une suite de sommets $s_0, s_1, \dots$ telle que pour tout i , la composante connexe de $G_{s_0, \dots, s_{i-1}}$ contenant s_i est infinie.
Initialisation. G est infini, donc on choisit s_0 quelconque.
Hérédité. Supposons avoir construit $s_0, \dots, s_i$ formant un chemin simple, avec s_i dans une composante infinie de $G \setminus \{s_0, \dots, s_{i-1}\}$. Par la question 5 appliquée à $G' =$ cette composante et $v = s_i$, il existe une composante infinie dans $G' \setminus \{s_i\}$, dont on choisit un sommet s_{i+1} voisin de s_i .
 La suite ainsi construite est un rayon de G .
7. ($\Rightarrow$) Si G est k -coloriable, toute restriction à un sous-graphe fini l'est aussi (la restriction d'une k -coloration est une k -coloration).
 ($\Leftarrow$) Supposons que tout sous-graphe fini de $G = (\mathbb{N}, E)$ soit k -coloriable. On construit une k -coloration de G par compacité (lemme de König ou argument diagonal).
 Pour chaque n , le sous-graphe induit $G[\{0, \dots, n\}]$ admet une k -coloration f_n . L'ensemble des k -colorations de $\{0, \dots, n\}$ est fini. Par le lemme de König (arbre finiment ramifié infini), il existe une suite infinie cohérente (f_n) ; sa limite définit une k -coloration de G tout entier.

Complexité arithmétique de la factorisation

On s'intéresse à la *complexité arithmétique* des algorithmes : chaque opération de base (addition, soustraction, multiplication, division euclidienne) sur des entiers coûte 1, quelle que soit la taille des opérandes. L'objectif est de montrer qu'il existe un algorithme de factorisation d'un entier $N = P \times Q$ (avec P, Q premiers, $N < 2^n$) utilisant $\mathcal{O}(n)$ opérations arithmétiques.

1. Donner un algorithme calculant N^K (pour N, K de taille n bits) en $\mathcal{O}(n)$ opérations arithmétiques.
2. Donner un algorithme calculant $\text{PGCD}(N, M)$ (pour N, M de taille n bits) en $\mathcal{O}(n)$ opérations arithmétiques. On justifiera la complexité.
3. En supposant l'existence d'un algorithme calculant $N!$ en $F(n)$ opérations arithmétiques, montrer qu'il existe un algorithme de factorisation de complexité $\mathcal{O}(n \times F(n))$. (On ne demande pas de preuve de correction, seulement une description.)
4. On s'intéresse au calcul d'un coefficient binomial $\binom{N}{M}$ pour $M \leq N$, tous deux de taille n bits. En utilisant l'identité

$$\forall k, \ell \in \mathbb{N}, \quad (2^\ell + 1)^k = \sum_{j=0}^k \binom{k}{j} 2^{\ell j},$$

montrer que ce calcul peut être effectué en $\mathcal{O}(n)$ opérations arithmétiques.

5. En utilisant la question précédente et l'identité $N! = \binom{N}{N/2} ((N/2)!)^2$ (valable pour N pair), montrer qu'on peut calculer $N!$ en $\mathcal{O}(n)$ opérations arithmétiques.
6. Montrer que tout algorithme calculant $N!$ requiert $\Omega(n)$ opérations arithmétiques, et conclure que l'algorithme de la question 5 est optimal à constante près.
7. En adaptant l'algorithme de la question 5, montrer qu'on peut calculer le plus petit entier i tel que $(2^i)! \not\equiv 0 \pmod{N}$ et $(2^{i+1})! \equiv 0 \pmod{N}$ en $\mathcal{O}(n)$ opérations arithmétiques.
8. Donner un algorithme de factorisation de N utilisant $\mathcal{O}(n)$ opérations arithmétiques.

Solution : Complexité arithmétique de la factorisation

1. Il s'agit de l'**exponentiation rapide** : on écrit K en binaire et on calcule N^K par la méthode des carrés successifs. Chaque bit de K donne lieu à au plus 2 opérations (un carré et une multiplication conditionnelle), et K a n bits, donc la complexité est $\mathcal{O}(n)$.
2. Il s'agit de l'**algorithme d'Euclide** $\text{PGCD}(N, M) = \text{PGCD}(M \bmod N, N)$.
Complexité. On suppose $N \leq M$. On a $\min(N, M \bmod N) \leq M/2$: en effet, $N + (M \bmod N) \leq M$ (puisque $N \leq M$), donc le plus petit des deux est $\leq M/2$. Ainsi, après deux itérations, la somme des tailles des deux nombres diminue d'au moins 1 bit. On atteint 0 en $\mathcal{O}(n)$ étapes, chacune coûtant $\mathcal{O}(1)$.
3. On cherche le plus petit i tel que $i! \bmod N = 0$; on a alors $i = \max(P, Q)$.
On effectue une **recherche dichotomique** de $i_0 = \max(P, Q)$ sur $[1, N]$. L'intervalle est de taille $N < 2^n$, donc la dichotomie requiert $\mathcal{O}(n)$ étapes. À chaque étape, on calcule $i! \bmod N$ en $F(n) + \mathcal{O}(1)$ opérations. La complexité totale est $\mathcal{O}(n \times F(n))$.
4. On implémente d'abord une opération de **sélection de bits** : extraire les bits de X situés entre les positions m_0 et m_1 coûte $\mathcal{O}(1)$ opérations (une fois les puissances de 2 précalculées par exponentiation rapide).
On prend $\ell = k = N$ dans l'identité, ce qui donne

$$(2^N + 1)^N = \sum_{j=0}^N \binom{N}{j} 2^{Nj}.$$

Le coefficient $\binom{N}{M}$ apparaît donc dans les bits de positions NM à $NM + N$ de $(2^N + 1)^N$. On calcule :

- (a) $2^N + 1$ en $\mathcal{O}(n)$ opérations (exponentiation rapide) ;
 - (b) $(2^N + 1)^N$ en $\mathcal{O}(n)$ opérations (exponentiation rapide) ;
 - (c) 2^{NM} et 2^{NM+N} en $\mathcal{O}(n)$ opérations ;
 - (d) la sélection en $\mathcal{O}(1)$.
5. On décrit un algorithme récursif calculant simultanément $K!$, $(2^N + 1)^K$ et 2^{NK} (avec N paramètre fixé, $K \leq N$).
— **K impair** : $K! = K \times (K-1)!$, $(2^N + 1)^K = (2^N + 1)^{K-1} (2^N + 1)$, $2^{NK} = 2^{N(K-1)} \cdot 2^N$. Appel récursif en $K-1$.
— **K pair** : $K! = \binom{K}{K/2} ((K/2)!)^2$, $(2^N + 1)^K = ((2^N + 1)^{K/2})^2$, $2^{NK} = (2^{N(K/2)})^2$. Appel récursif en $K/2$, puis 2 opérations par calcul, puis sélection pour $\binom{K}{K/2}$, puis une multiplication pour $K!$.

On appelle ensuite avec $K = N$. La profondeur de récursion est $\mathcal{O}(n)$ (à chaque appel pair on divise par 2, à chaque appel impair on réduit de 1, et les appels impairs consécutifs sont au plus en nombre constant entre deux divisions). Chaque niveau coûte $\mathcal{O}(1)$ opérations, soit $\mathcal{O}(n)$ au total.

6. Tout algorithme effectuant c opérations ne peut produire que des entiers inférieurs à $(2^n)^{2^c}$. Or, par la formule de Stirling, $N! = \Theta(\sqrt{2\pi N} (N/e)^N)$, qui a $\Theta(N \log N) = \Theta(n \cdot 2^n)$ bits. On a donc besoin d'au moins $\Omega(n)$ opérations.
7. Soit i le plus petit entier tel que $2^i > N$. On appelle la fonction récursive de la question 5 pour calculer $(2^i)!$, en **mémorisant** chaque valeur de sortie $(2^j)!$ pour $1 \leq j \leq i$. On calcule alors $(2^j)! \bmod N$ pour chaque j en une division euclidienne supplémentaire. Le surcoût est $\mathcal{O}(n)$ divisions, soit $\mathcal{O}(n)$ opérations au total.
8. Par la question 7, trouver i_0 tel que $(2^{i_0})! \not\equiv 0$ et $(2^{i_0+1})! \equiv 0 \pmod{N}$.
Si $\text{PGCD}((2^{i_0})!, N) > 1$, on obtient directement un facteur non trivial de N .
Sinon, effectuer une recherche dichotomique du plus petit entier pair I dans $[2^{i_0}, 2^{i_0+1}]$ tel que $I! \equiv 0 \pmod{N}$, en utilisant $I! = \binom{I}{I/2} ((I/2)!)^2$ et en maintenant $(2^N + 1)^I$ par produits successifs avec les puissances précalculées $(2^N + 1)^{2^j}$. Chaque étape de dichotomie coûte $\mathcal{O}(1)$ opérations après précalcul et il y en a $\mathcal{O}(n)$.
Une fois $I = \max(P, Q)$ trouvé, calculer $\text{PGCD}(I, N)$ en $\mathcal{O}(n)$ opérations pour obtenir P et Q .
La complexité totale est $\mathcal{O}(n)$ opérations arithmétiques.

Inégalités et propriétés des degrés dans un graphe

Soit $G = (S, A)$ un graphe non orienté d'ordre n (nombre de sommets) et de taille $|A|$ (nombre d'arêtes). On note $\deg(s)$ le degré d'un sommet $s \in S$.

1. Compléter et démontrer la formule des sommets : $\sum_{s \in S} \deg(s) = \dots$. En déduire le nombre d'arêtes d'un arbre à n sommets. Montrer qu'un arbre possédant un sommet de degré 2017 possède au moins 2017 feuilles.
2. Quel est le nombre minimal d'arêtes que peut avoir un graphe connexe à n sommets? Soit $k \in \mathbb{N}$. Quel est le nombre minimum de composantes connexes d'un graphe à n sommets et $n - k$ arêtes?
3. Montrer que dans tout graphe avec au moins 2 sommets ($n \geq 2$), il existe au moins deux sommets de même degré.
4. Soit G un graphe dont tous les sommets sont de degré 3. Montrer que n est nécessairement pair. Pour $n \geq 2$, construire un graphe à $2n$ sommets dont tous les sommets sont de degré 3.
5. Soit G un graphe d'ordre $2p$. On suppose que pour tout $s \in S$, $\deg(s) \geq p$. Démontrer que ce graphe est nécessairement connexe.
6. Soit $G = (S, A)$ un graphe **sans triangle** à n sommets.
 - (a) Montrer que pour toute arête $(x, y) \in A$, on a $\deg(x) + \deg(y) \leq n$.
 - (b) Montrer que $\sum_{x \in S} \deg(x)^2 = \sum_{(x, y) \in A} (\deg(x) + \deg(y))$.
 - (c) En déduire que $\sum_{x \in S} \deg(x)^2 \leq |A| \cdot n$.
 - (d) À l'aide de l'inégalité de Cauchy-Schwarz, montrer que $(\sum_{x \in S} \deg(x))^2 \leq n \cdot \sum_{x \in S} \deg(x)^2$.
 - (e) En combinant les résultats précédents, démontrer le théorème de Mantel : $|A| \leq \frac{n^2}{4}$.
7. Rappeler la définition de la matrice d'adjacence M d'un graphe. Que représente la matrice M^i ? Comment utiliser cette information pour déterminer si G contient un triangle (un cycle de longueur 3)? Trouver un vecteur colonne X non nul tel que si G est k -régulier (tous les sommets de degré k), alors $MX = kX$.

Solution : Inégalités et propriétés des degrés dans un graphe

1. $\sum_{s \in S} \deg(s) = 2|A|$ (chaque arête est comptée deux fois). Pour un arbre, $|A| = n - 1$.
2. Un graphe connexe a au moins $n - 1$ arêtes. Un graphe à n sommets et $n - k$ arêtes a au moins k composantes connexes.
3. Les degrés possibles sont dans $\llbracket 0, n - 2 \rrbracket$ ou $\llbracket 1, n - 1 \rrbracket$. Dans les deux cas, n sommets pour $n - 1$ valeurs possibles impose une répétition.
4. $2|A| = \sum \deg(s) = 3n$, donc $3n$ est pair, donc n est pair. Construction : un cycle C_{2n} avec des cordes reliant les sommets opposés.
5. Par l'absurde, si deux sommets u, v ne sont pas reliés, leurs voisinages respectifs $N(u)$ et $N(v)$ sont disjoints et ne contiennent ni u ni v . Alors $n \geq |N(u)| + |N(v)| + |\{u, v\}| \geq p + p + 2 = 2p + 2$, ce qui contredit $n = 2p$.
6. **Théorème de Mantel :**
 - (a) Si $\deg(x) + \deg(y) > n$, x et y auraient un voisin commun z (par le principe des tiroirs), formant un triangle (x, y, z) .
 - (b) En sommant $\deg(x)$ pour chaque sommet, chaque $\deg(x)^2$ apparaît car x est compté dans $\deg(x)$ arêtes.
 - (c) $\sum \deg(x)^2 = \sum_{(x,y) \in A} (\deg(x) + \deg(y)) \leq \sum_{(x,y) \in A} n = |A|n$.
 - (d) $(\sum \deg(x))^2 \leq n \sum \deg(x)^2$ est l'application directe de CS avec le vecteur $(1, \dots, 1)$.
 - (e) $(2|A|)^2 \leq n(|A|n) \implies 4|A|^2 \leq |A|n^2 \implies |A| \leq \frac{n^2}{4}$.
7. $(M^i)_{j,k}$ est le nombre de chemins de longueur i entre j et k . G a un triangle ssi $\text{Tr}(M^3) > 0$. Pour le vecteur propre, $X = (1, 1, \dots, 1)^T$.

Demain il pleut

Étant donné une matrice 2D W de taille $m \times n$ qui représente une carte constituées de cases de terre et d'eau ($W_{i,j} = 0$ si $W_{i,j}$ est de la terre, sinon $W_{i,j} = 1$), on cherche à assigner une hauteur à chaque case pour obtenir le plus haut pic possible sur la carte. L'assignation h doit vérifier :

- $h_{i,j} \geq 0$
- La hauteur d'une case d'eau est de 0
- La différence de hauteur entre deux cases qui partagent un côté ne peut pas être plus grande que 1.

Renvoyez la hauteur maximale. Quelle est la complexité de votre algorithme ?