

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE  
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE

جامعة بجاية

Université de Béjaïa

Tasdawit n'Bgayet

UNIVERSITÉ ABDERRAHMANE MIRA DE BÉJAIA

FACULTÉ DES SCIENCES EXACTES

DÉPARTEMENT DE RECHERCHE OPÉRATIONNELLE

# MÉMOIRE DE FIN D'ÉTUDES

En vue de l'obtention du diplôme de

**MASTER en Recherche Opérationnelle**

**Option : Modélisation Mathématique et Technique de décision**

---

THÈME

Parité et forte parité de l'Indice Chromatique de  
nouvelles classes d'arbres.

---

Présenté par :

M<sup>elle</sup> **BOUZABAL Sarah**  
M<sup>elle</sup> **SEDDIKI Douniazad**

Examiné par :

M<sup>r</sup> **S. TAQUINET**

M<sup>r</sup> **DJ. TALEM**

M<sup>me</sup> **L. YOUNSI**

Encadré :

M<sup>r</sup> **K. KABYL**

Année Universitaire 2020 — 2021

# **REMERCIEMENTS**

*Nous remercions d'abord et avant tout le bon dieu tout puissant de nous avoir donné le courage et la force pour réaliser ce travail.*

*Nous désirons remercier notre promoteur Mr KABYL Kamal pour sa disponibilité, son aide précieuse et ses conseils qui nous ont accompagné tout au long de ce travail. Nous sommes très reconnaissantes d'avoir cru en nos capacités et d'avoir accordé sa confiance, et aussi pour ses qualités scientifiques, pédagogiques et humaine.*

*Nous remercions les membres du jury qui ont accepté d'évaluer notre travail.*

*Nous remercions aussi nos parents et proches pour leurs aides et encouragements.*

*Sans oublier nos amis et camarades pour leurs soutiens qu'ils nous ont apportés.*

*Merci pour tout.*

## *Dédicace*

*Du profond de mon coeur je dédie ce travail à tous ceux qui me sont chers :*

*À la mémoire de mon grand-père*

*Ce travail est dédié à lui, qui est décidé trop tôt, qui m'a toujours poussé et motivée dans mes études.*

*À ma très chère grande mère*

*Bien aimée que dieu vous garde pour nous.*

*À la femme*

*Qui a souffert sans me laisser souffrir qui n'a jamais dit non à mes exigences et qui n'a épargné aucun effort pour me rendre heureuse. aucun dédicace ne saurait exprimer mon respect, mon amour éternel et ma considération pour les sacrifices que vous avez consenti pour mon instruction et mon bien être. je t'aime yemma.*

*À l'homme*

*Mon précieux offre du bien qui doit ma vie ma réussite et tout mon respect mon cher père.*

*À mes tantes Hayette et Djahida.*

*À mon adorable petite soeur.*

*Lamis qui sait toujours comment procurer la joie et le bonheur pour tous la famille.*

*À mes frères Larbi et Fares .*

*À mon cher pour son encouragement et toute sa famille.*

*À ma famille et mes proches.*

*À ma meilleure amie ma binôme pour son soutien moral, sa patience et sa compréhension tout au long de ce projet et toute sa famille.*

*À mes meilleurs amis Sabrina, Siham, Wissam, Firouz, Nassima, Souad, Zahra, Kamy.*

*À groupe des fous {Cylia, Samia, Farouk, Wali, Fares, Mouh}.*

*Qui m'a beaucoup aidé à accomplir ce modeste travail.*

*Merci*

***BOUZABAL Sarah***

## *Dédicace*

*Le présent travail n'aurait pas abouti sans le bienveillant soutien de certaines personnes. Je voudrais les prier d'accueillir ici tous mes sentiments de gratitude.*

**Je** dédie ce travail :

**A** ma très chère mère, aucune dédicace ne saurait être à la hauteur pour exprimer ce que tu mérites pour tous les sacrifices que tu n'as cessé de me donner depuis mon plus jeune âge, et même à l'âge adulte.

*Je te dédie ce travail en témoignage de mon profond amour. Puisse Dieux, le tout puissant, te préserver et t'accorder santé, longue vie et bonheur.*

**Au** meilleur des pères, les mots ne suffisent guère pour exprimer l'attachement et le respect que je te porte. Ce travail est le fruit de tes sacrifices que tu as consentis pour mon éducation et ma formation.

**A** mes soeurs Hanane, Wissam et Sabrina ainsi leurs petites familles.

**A** mes frères Ziad, Hamza et Bilal ainsi leurs petites familles.

**A** toute ma famille mes cousines et cousins.

**A** mes plus chères amies Asma, Hayette et Kamelia.

**A** mon meilleur ami Toufik.

**A** mon adorée et ma binôme Sarah et toute sa famille.

**A** la mémoire de mon grand père.

**SEDDIKI Douniazad**

# Table des matières

<b>Table des figures</b>	<b>7</b>
<b>Introduction</b>	<b>9</b>
<b>1 Concepts fondamentaux des graphes</b>	<b>11</b>
1.1 Introduction	11
1.2 Graphe	11
1.3 Quelques types de graphe	12
1.3.1 Graphe simple :	12
1.3.2 Sous graphe	12
1.3.3 Graphe partiel	12
1.3.4 Sous graphe partiel	12
1.4 Chaînes et cycles	12
1.4.1 Chaîne et cycles hamiltoniens	13
1.4.2 Longueur de chaînes et de cycles	13
1.5 Stable et clique	13
1.5.1 Stable	13
1.5.2 Clique	14
1.5.3 Couplage parfait(complet )	14
1.6 Notion de degré	15
1.6.1 Notation	16
1.6.2 Formule des degrés	16
1.7 Opérations sur les graphes	16
1.7.1 Somme cartésienne	16
1.7.2 Produit cartésien	16
1.7.3 Subdivisions de graphes	16
1.7.4 Homomorphisme	17
1.7.5 Isomorphisme	17
1.8 Notion de connexité dans les graphes	17
1.8.1 La connexité	17
1.8.2 La composante connexe	17
1.9 Quelques graphes particuliers	18
1.9.1 Graphe complet	18
1.9.2 Graphe biparti	18
1.9.3 Graphe biparti complet	19
1.9.4 Graphe biparti équilibré	19

1.9.5	Arbre et arborexence . . . . .	20
1.9.6	Graphe régulier . . . . .	20
1.9.7	Graphe de Hamming . . . . .	20
1.9.8	Graphe planaire . . . . .	21
1.9.9	Hypercube . . . . .	21
1.10	Représentations d'un graphe . . . . .	23
1.10.1	Liste d'adjacences . . . . .	23
1.10.2	Matrice d'adjacences . . . . .	23
1.11	Conclusion . . . . .	24
<b>2</b>	<b>Coloration des graphes et coloration des arêtes</b>	<b>25</b>
2.1	Introduction . . . . .	25
2.2	Coloration propre . . . . .	25
2.3	Coloration des sommets . . . . .	25
2.3.1	Algorithme . . . . .	26
2.3.2	Nombre chromatique . . . . .	26
2.4	Coloration des arêtes . . . . .	27
2.4.1	Indice chromatique . . . . .	28
2.4.2	Encadrement de l'indice chromatique . . . . .	28
2.4.3	Propriétés . . . . .	29
2.5	Parité et forte parité de quelques arbres binaires . . . . .	32
<b>3</b>	<b>Parité et forte parité de l'Indice Chromatique de deux nouvelles classes d'arbres et application</b>	<b>36</b>
3.1	Introduction . . . . .	36
3.2	Caractérisation de deux nouvelles classes d'arbres . . . . .	36
3.2.1	Arbre $\hat{B}_n$ . . . . .	36
3.2.2	Arbre $T_n$ . . . . .	41
3.3	Application dans Code Blocks . . . . .	43
	<b>Conclusion</b>	<b>48</b>
	Bibliographie . . . . .	49

# Table des figures

1.1	$G_A$ est le sous graphe de $G$ . . . . .	12
1.2	$G_v$ graphe partiel de $G$ . . . . .	13
1.3	Graphe $K_4$ . . . . .	19
1.4	Graphe biparti à 5 sommets et 5 arêtes. . . . .	19
1.5	Graphe $K_{3,3}$ . . . . .	19
1.6	Graphe 3-régulier . . . . .	20
1.7	Graphe planaire . . . . .	21
1.8	les quatres premiers hypercubes . . . . .	22
1.9	Graphe à 6 sommets et 11 arcs et sa liste d'adjacence associée. . . . .	23
1.10	Graphe à 4 sommets et 4 arêtes et la matrice d'adjacence associée. . . . .	23
2.1	Coloration des arêtes . . . . .	27
2.2	Graphe adjoint de $G$ . . . . .	27
2.3	3-coloration de $L(G)$ . . . . .	28
2.4	Indice Chromatique de l'hypercube $Q_2$ . . . . .	28
2.5	Graphe $K_{1,3}$ . . . . .	29
2.6	Indice chromatique fort. . . . .	30
2.7	Graphe planaire de degré 7. . . . .	30
2.8	Arbre $T$ et son indice chromatique $P(G)$ . . . . .	31
2.9	$\hat{P}(Q_3) = 3$ . . . . .	31
2.10	Arbres $D_1, D_2, D_3$ . . . . .	32
2.11	$\hat{P}(D_1)$ et $\hat{P}(D_2)$ . . . . .	32
2.12	Plongement de l'arbre $D_2$ dans $Q_4$ . . . . .	33
2.13	Arbre $B_2$ et arbre $B_3$ . . . . .	34
2.14	$P(B_2)$ et $P(B_3)$ . . . . .	34
2.15	Plongement de l'arbre $B_2$ dans $Q_3$ . . . . .	35
3.1	Arbres $\hat{B}_2$ et $\hat{B}_3$ . . . . .	37
3.2	Coloration des arêtes de l'arbre $\hat{B}_2$ , $\alpha(\hat{B}_2) = 3$ . . . . .	38
3.3	Les permutations possible de l'arbre $\hat{B}_2$ . . . . .	38
3.4	Coloration des arêtes des arbres $\hat{B}_n$ . . . . .	39
3.5	Coloration des arêtes des arbres $\hat{B}_3$ et $\hat{B}_4$ . . . . .	40
3.6	Arbre $T_1$ . . . . .	41
3.7	Arbre $T_2$ . . . . .	41
3.8	Arbre $T_3$ . . . . .	42
3.9	Coloration des arêtes des arbres $T_1$ et $T_2$ . . . . .	42
3.10	Coloration des arêtes de l'arbre $T_3$ . . . . .	43

3.11	Coloration des arêtes de l'arbre $T_n$ .	43
3.12	l'interface de code blocks.	44
3.13	Résultat.	46
3.14	Arbre couvrant de poids minimum.	47



# Introduction générale

L'histoire de la théorie des graphes débute avec les travaux d'Euler au 18ème siècle et trouve son origine dans l'étude de certains problèmes. La théorie des graphes s'est alors développée dans diverses disciplines telles que la chimie, la biologie, les sciences sociales (réseaux de transports), gestion de projets, informatique (topologie des réseaux, complexité algorithmique, protocoles de transferts), la physique quantique, etc. Depuis le début du 20ème siècle, elle constitue une branche à part entière des mathématiques, grâce aux travaux de König, Menger, Cayley puis de Berge et d'Erdős. Cette branche des mathématiques a connu un grand regain d'intérêt suite à l'émergence des réseaux sociaux Internet.

De manière générale, un graphe permet de représenter la structure, les connexions d'un ensemble complexe en exprimant les relations entre ses éléments : réseau de communication, réseaux routiers, interaction de diverses espèces animales, circuits électriques. Les graphes constituent donc une méthode de pensée qui permet de modéliser une grande variété de problèmes en les ramenant à l'étude de sommets et d'arcs.

Le problème de coloration de graphes est l'un des origines de la théorie des graphes, ce problème remonte au 19ème siècle lorsque Guthrie a remarqué que quatre couleurs étaient suffisantes pour colorer la carte d'Angleterre, sans donner la même couleur à deux régions ayant une frontière commune. Il pose alors la question de savoir si quatre couleurs suffisent toujours pour colorier n'importe quelle carte géographique de sorte que deux régions voisines n'aient pas la même couleur. Malgré un énoncé simple, cette conjecture est restée non résolue pendant plus d'un siècle. Il fallut attendre 1978 pour qu'Appel et Haken parviennent à démontrer ce résultat à l'aide d'un ordinateur.

De nombreuses variantes et extensions de colorations de graphes ont été considérées dans la littérature. Les colorations impropres (également connues sous le nom de colorations généralisées ou relaxées) sont une généralisation des colorations propres où la contrainte consistant à colorier de façon différente deux sommets adjacents est supprimée.

Voici un résumé des différents chapitres qui composent ce travail :

Le premier chapitre est consacré aux notions de base, nous donnons des définitions fondamentales et rappels sur les notions de base de la théorie des graphes, que nous utiliserons aux cours de ce mémoire.

Le deuxième chapitre porte sur la coloration des arêtes de graphes. Nous parlerons

de graphe avec ses différentes manières ainsi que la caractérisation et la coloration de quelques classes de graphes à savoir les graphes planaires et les graphes bipartis. Comme nous citons la notion de parité et forte parité de quelques arbres binaires.

Dans le troisième chapitre, nous présentons deux nouvelles classes d'arbre pour lesquelles nous avons indiqué la parité et la forte parité de l'indice chromatique, avec l'implémentation de l'algorithme de prime en utilisant le code block  $c^{++}$ .

Nous terminons notre travail par une conclusion générale, dans laquelle nous résumons tout ce que nous avons fait dans ce mémoire.

# 1

## Concepts fondamentaux des graphes

### 1.1 Introduction

Un bon dessin vaut mieux qu'un long discours. Le langage des graphes essaie de mettre en pratique cette idée. Bien souvent, en effet, c'est un réflexe naturel qui pousse à abstraire une situation donnée en traçant, sur une feuille de papier, des points pouvant représenter des individus, des localités, des corps chimiques,... relié entre eux par des lignes ou des flèches symbolise une certaine relation. Dont on trouve les outils mathématiques et les algorithmes mis au point en théorie des graphes permettent de résoudre une multitude de problèmes, tels que les problèmes de cheminement, d'ordonnancement, d'affectation, etc... L'objectif de ce chapitre est de rappeler quelques notions de bases et notations couramment utilisées dans la théorie des graphes qui seront utilisées le long de ce manuscrit, pour le reste des notations nous adopterons la terminologie de [7, 9, 10].

### 1.2 Graphe

Un graphe  $G$  est un objet mathématique composé d'un ensemble  $V$  non vide et fini de points  $\{v_1, v_2, \dots, v_n\}$  appelés sommets (noeuds), et par un ensemble des couples de sommets noté  $E$  (qui peut être vide) de segments (flèches)  $\{e_1, e_2, \dots, e_m\}$  appelés arêtes (arcs), en reliant chaque paires de sommets  $v_1$  et  $v_2$  par une arête (arc)  $e$  s'écrit comme étant  $e = \{v_1, v_2\}$  ou  $(v_1 v_2)$ , les points  $v_1$  et  $v_2$  appelés les extrémités de  $e$ . [9]

On appelle ordre d'un graphe  $G$  le nombre de ses sommets, noté par :  $|V(G)|$

Soit  $x, y \in V$ , on dit que  $x$  est adjacent à  $y$  si  $x$  et  $y$  sont reliés par un arête (arc).

Si  $e = \{x, y\}$  est une arête, et si  $x = y$  alors  $e$  définit une boucle. en d'autre terme une boucle est une arête reliant un sommet à lui même. [9]

Le graphe  $H = (V, E')$  est un graphe partiel de  $G$ , si  $E' \subseteq E$ , Autrement dit on obtient  $H$  en enlevant une ou plusieurs arêtes au graphe  $G$ . [3]

Un sous graphe engendré par  $A \subseteq V$  est un graphe dont les sommets sont ceux de  $A$  et

les arêtes sont celles ayant les deux extrémités dans  $A$ . on le note  $G_A$ . [10]

## 1.3 Quelques types de graphe

### 1.3.1 Graphe simple :

Un graphe  $G$  est dit simple si tous ses sommets sont sans boucles et entre chaque paire de sommets, il y a au plus une arête.

### 1.3.2 Sous graphe

Un sous graphe de  $G = (V, E)$  engendré par l'ensemble de sommets  $A \subseteq V$ , est le graphe :

$G_A = (A, E_A)$  où  $E_A = \{e = \{x, y\} \in E / x \in A \text{ et } y \in A\}$ , la figure 1.1 montre un graphe  $G$  et son sous graphe  $G_A$ .

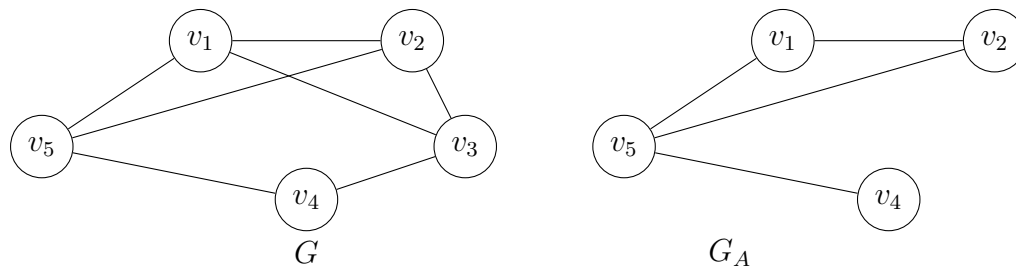


FIGURE 1.1 –  $G_A$  est le sous graphe de  $G$

avec,  $E_A = \{e_1, e_2, e_4, e_5\}$  et  $A = \{v_1, v_2, v_4, v_5\}$

### 1.3.3 Graphe partiel

Un graphe partiel de  $G$  engendré par l'ensemble d'arcs (resp. d'arêtes)  $V$  est le graphe  $G_V = (X, V)$ . La figure 1.2 montre un graphe  $G$  et son graphe partiel.

### 1.3.4 Sous graphe partiel

Un sous graphe partiel de  $G$  engendré par l'ensemble de sommets  $A$  et l'ensemble d'arcs (resp. d'arêtes)  $V$  est le graphe  $G_{A,V} = (A, V_A)$ .  $V_A$  est l'ensemble d'arcs (resp. arêtes) qui ont leurs deux extrémités dans le sous ensemble  $V$ .

## 1.4 Chaînes et cycles

On appelle chaîne entre deux sommets  $x$  et  $y$  d'un graphe  $G$ , une suite de sommets  $x_1, \dots, x_k$  dont deux consécutifs sont adjacents, avec  $x_1 = x$  et  $x_k = y$ . Une chaîne qui n'utilise pas deux fois le même sommet est dite **élémentaire**, une chaîne qui n'utilise

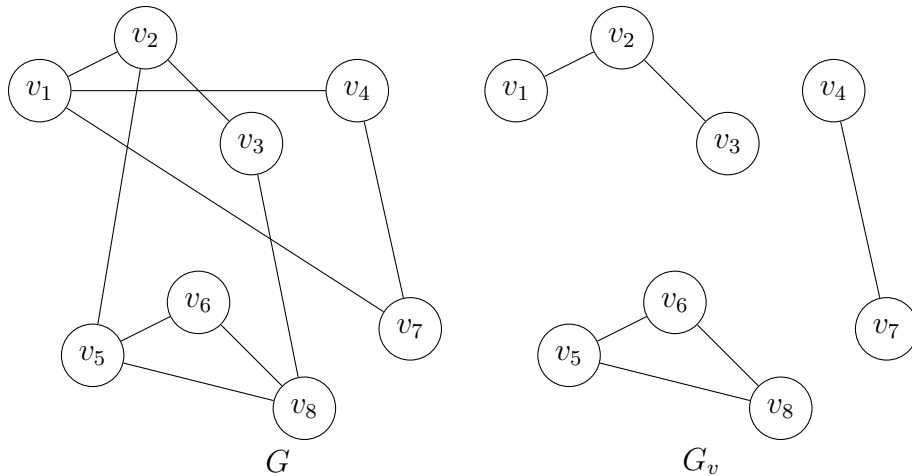


FIGURE 1.2 –  $G_v$  graphe partiel de  $G$

pas deux fois la même arête est dite **simple**. Une chaîne élémentaire est donc une chaîne simple.

On appelle **cycle** dans un graphe, une chaîne simple dont les extrémités initiale et finale sont confondues. Dans la suite, sans spécifications contraires, on parlera seulement de chaîne et de cycle élémentaires.

### 1.4.1 Chaîne et cycles hamiltoniens

On appelle chaîne (resp.cycle) **hamiltonienne** (resp.**hamiltonien**) une chaîne (resp.cycle) passant, une fois et une seule, par tous les sommets d'un graphe  $G$ . Un graphe  $G$  qui possède un cycle hamiltonien est dit **graphe hamiltonien**.

### 1.4.2 Longueur de chaînes et de cycles

La Longueur d'une chaîne (resp.cycle) est le nombre d'arêtes de cette chaîne (resp.cycle) et on note  $P_n$  (resp.  $C_n$ ) la chaîne élémentaire à  $n$  sommets, de longueur  $n - 1$  (resp.  $n$ ).[\[18\]](#)

## 1.5 Stable et clique

### 1.5.1 Stable

- Un stable  $S$  dans un graphe  $G$  est un sous-ensemble de sommets de  $G$  tel que  $G[S]$  est un graphe sans arêtes.
- Un stable maximal au sens de l'inclusion des ensembles.
- Un stable de cardinalité maximale est dit stable maximum.
- Un stable maximum est obligatoirement maximal.
- Un stable maximal n'est pas forcément maximum.

## 1.5.2 Clique

- Une clique est un sous-ensemble de sommets qui induit un sous graphe complet et symétrique.
- Une clique d'ordre  $n$  sommets possède  $n(n - 1)/2$  arêtes.
- On note  $K_n$  l'ensemble des cliques d'ordre  $n$  d'un graphe  $G$ .
- Une clique  $C$  est dite maximale s'il n'existe pas une autre clique  $C'$  qui la contient. On note  $\mathcal{T}_G$  l'ensemble des cliques maximales de  $G$ .

## 1.5.3 Couplage parfait (complet )

Un couplage  $M$  du graphe  $G$  est dit parfait si tout sommet du graphe  $G$  est incident exactement à une arête de couplage  $M$ .

**Remarque:**

- Si tous les sommets de  $G$  sont  $M$ -saturés, alors on dit que le couplage  $M$  est parfait.
- On peut pas avoir un couplage parfait si un cardinal du  $G$  est impaire

### Couplage maximal

Couplage maximal est un couplage  $M$  dans lequel toute arête du graphe possède au moins une extrémité commune avec une arête de couplage  $M$ . c'est à dire :

$$\forall e \in E \text{ et } e \notin M \text{ on a } M \cup e \text{ n'est plus un couplage de } G .$$

**Remarque:** Le couplage  $M$  est maximal s'il n'existe pas de couplage qui le contient.

### Couplage maximum :

Ce type de couplage vise à avoir le plus grand nombre d'arêtes possible, il peut y avoir plusieurs couplages maximums dans un seul graphe.

**Théorème 1.5.1.** *Un couplage  $M$  de  $G$  est maximum si et seulement s'il n'existe pas de chaîne alternée  $M$  – augmentante.*[8]

---

**Algorithm 1** Algorithme de recherche d'une chaîne augmentante

---

**Début :**

1. Soit  $\{F\} \leftarrow \emptyset, \forall v \in V$  un ensemble de sommets non-marqué, et  $M$  un couplage de  $G$  seuls les  $e \in M$  qui sont marquées .
2. Pour tout  $v \in V$  non-couvert par le couplage  $M$  faire : • Créer un arbre  $T - \{v\}$  composé uniquement d'un seul sommet de  $V$  ,

$$F \leftarrow F \cup T - v.$$

3. **Tant que** il existe des sommets  $v$  appartenant à  $F$  non-marqués externes faire :  
**Tant que** il existe une arête  $e = (v, w)$  non-marquée de  $F$  faire : **Si**  $w \in F$  **Alors** soit  $x$  le sommet appariée à  $w$  donc :ajouter  $(v, w)$  et  $(w, x)$  à l'arbre contenant  $v$   
**Sinon** retourner à l'étape (2). marquer  $e$  et  $v$ ; (ne plus les traiter)

**return**  $F$ . **Fin.**

---

**L'algorithme d'Edmonds :**

Cet algorithme était créé en 1965 par Jack Edmonds, nommé aussi algorithme de Blossom, permettant de déterminer un couplage maximum dans un graphe  $G$ , son principe procède en marquant les sommets et les arêtes par des étiquettes cela fait appel à l'algorithme de recherche d'une chaîne augmentante (d'après le théorème de Berge).

---

**Algorithm 2** L'algorithme de Couplage- $Max(G, M)$

---

**Début :**

1. Entrées un graphe  $G = (S, E)$ , un couplage  $M$ .
  2. Faire appel à l'algorithme de recherche d'une chaîne augmentante.  
**Si** il existe de chaîne augmentant **Alors** :  
retourner  $M$  ;  
**Sinon** retourner  $G, M$  (modifier par la chaîne augmentante).  
**Fin.**
- 

## 1.6 Notion de degré

**Définition 1.6.1.** *Soit  $G = (V, E)$  un graphe non orienté (resp.  $G = (V, U)$  un graphe orienté). A tout sommet  $v \in V$ , on peut associer une valeur entière positive ou nulle, notée  $d_G(v)$ , qu'on appelle degré du sommet  $v$ . Cette valeur est définie comme suit :  $d_G(v)$  = nombre de fois où  $v$  est extrémité d'un arc (resp. d'une arête).*

**Remarque:** Pour tout graphe, nous avons :  $d_G(v) \geq |V(v)|$ . Si  $G$  est un graphe non orienté simple alors on a :  $d_G(v) = |V(v)|$ .

### 1.6.1 Notation

- On appelle degré minimal d'un graphe  $G$  qu'on note par  $\delta(G)$ , le plus petit degré dans le graphe  $G$ .  $\delta(G) = \min\{d_G(v)\}$
- On appelle degré maximal d'un graphe  $G$  qu'on note par  $\Delta(G)$ , le plus grand degré dans le graphe  $G$ .  $\Delta(G) = \max\{d_G(v)\}$
- Si  $d_G(v) = 0$ ; Alors  $v$  est dit sommet isolé.
- Si  $d_G(v) = 1$ ; Alors  $v$  est dit sommet pendant.
- Un arc (resp. Une arête) incident(e) à un sommet pendant est aussi dit(e) arc (arête) pendant(e).

### 1.6.2 Formule des degrés

Pour tout graphe non orienté  $G = (V, E)$ , on a :

$$\sum_{v \in V} d_G(v) = 2|E|$$

## 1.7 Opérations sur les graphes

Plusieurs opérations courantes permettent de construire à partir de deux graphes d'autres graphes, Commençons par les plus simples .

### 1.7.1 Somme cartésienne

Étant donnés deux graphes  $G = (S, E)$  et  $H = (V, A)$ , la somme cartésienne de  $G$  avec  $H$ , notée  $G \square H$ , est le graphe défini sur l'ensemble de sommets  $S \times A$  tel que deux sommets  $(u, u')$  et  $(v, v')$  sont adjacents si et seulement si  $(u = v \text{ et } u'v' \in A \text{ où } (u' = v' \text{ et } uv \in E))$ .

Il est à noter que le graphe  $G \square H$  possède  $|S| \times |V|$  sommets et  $|S| \times |A| + |V| \times |E|$  arêtes.

### 1.7.2 Produit cartésien

Étant donnés deux graphes  $G = (S, E)$  et  $H = (V, A)$ , le produit Cartésien de  $G$  par  $H$ , noté  $G \times H$ , est le graphe défini sur l'ensemble de sommets  $S \times V$  tel que deux sommets  $(u, u')$  et  $(v, v')$  sont adjacents si et seulement si  $uv \in E$  et  $u'v' \in A$ .

### 1.7.3 Subdivisions de graphes

En remplaçant les arêtes d'un graphe  $G = (S, E)$  par des chaînes disjointes intérieurement, on obtient une subdivision de  $G$ .

Une subdivision d'une arête  $uv$  du graphe  $G$  est le remplacement de cette arête par une chaîne  $u = u_0, u_1, \dots, u_{p-1}, u_p = v$ , où les  $u_i$  sont les nouveaux sommets insérés sur l'arête  $uv$  pour tout  $i \in \{1, 2, \dots, p-1\}$ .



## 1.7.4 Homomorphisme

La notion d'homomorphisme s'adresse à tous les graphes simples ou non, avec ou sans boucle.

Soient  $G = (S, E)$  et  $H = (V, A)$  deux graphes, une application  $f : S \rightarrow V$  qui envoie les sommets de graphe  $G$  sur ceux de graphe  $H$  est dite morphisme si et seulement si l'image de toute arête de graphe  $G$  est une arête de graphe  $H$ , c'est à dire :

$$\forall (u, v) \in S \times S, uv \in E \implies f(u)f(v) \in A$$

.

### Remarque:

- S'il existe un morphisme à la fois de graphe  $G$  dans le graphe  $H$ , et un de graphe  $H$  dans le graphe  $G$ , on note que les graphes  $G$  et  $H$  sont homomorphiquement équivalents.
- Un endomorphisme est un morphisme d'un graphe en lui-même.

## 1.7.5 Isomorphisme

Deux graphes  $G = (S, E)$  et  $H = (V, A)$  sont dits isomorphes si et seulement s'il existe une application bijective  $f : E \implies A$  qui vérifie la condition suivante :

$$uv \in E \Leftrightarrow f(u)f(v) \in A$$

Un isomorphisme de  $G$  dans lui même est appelé isomorphisme intérieur ou bien automorphisme, ceci signifie aussi que  $f$  est un morphisme et  $f^{-1}$  est un morphisme.

## 1.8 Notion de connexité dans les graphes

### 1.8.1 La connexité

Un graphe connexe est un graphe tel que pour toute paire de sommets  $x$  et  $y$ , il existe une chaîne reliant  $x$  et  $y$ , un graphe non connexe est décomposé en plusieurs composantes connexes.

### 1.8.2 La composante connexe

On appelle composantes connexes un ensemble de sommets, qui on deux à deux la relation de connexité, de plus tout sommet en dehors de la composante n'a pas de relation de connexité avec les sommets de cette composante.

- Pour la recherche des composants connexes d'un graphe nous utiliserons l'algorithme de marquage suivant.

### Algorithme de marquage

L'algorithme de marquage simple est un algorithme qui permet de déterminer les composantes connexe d'un graphe.[25][4]

## Principe

Soit  $G = (V, E)$  un graphe orienté (diagraphe). L'idée de cet algorithme est la suivante : pour un sommet quelconque  $v \in G$ , il s'agit de trouver toutes les chaînes reliant ce sommet aux autres sommets du graphe ; ainsi les sommets reliés au sommet  $v$  par une chaîne forment la composante connexe qui contient le sommet  $v$ .

**Enoncé :** \*Données : un graphe  $G = (V, E)$

---

**Algorithm 3** : Algorithme de marquage

---

1. Initialisation  $k \leftarrow 0$ ,  $A \leftarrow V$ .
  2. (a) Choisir un sommet  $v_i$  de  $A$  et le marquer avec un signe (+), puis marquer tous ses voisins avec le même signe. Continuer cette procédure jusqu'à ce qu'on ne puisse plus marquer de sommets.  
  
(b) Poser  $k = k + 1$  et  $C_k$  l'ensemble des sommets marqués ;  
(c) Retirer de  $A$  les sommets de  $C_k$  et poser  $A = A - C_k$  ;  
(d) Tester si  $A = \emptyset$  ;
    - Si oui terminer ; aller à (3)
    - Si non aller à (2) ;
  3. Le nombre de composantes connexes de  $(G)$  est  $k$ . Chaque ensemble  $C_i$ ,  $i = 1, \dots, k$  correspond aux sommets d'une composante connexe de  $(G)$ .
- 

\*Résultat : le nombre  $k$  de composantes connexes de  $G$  ainsi que la liste  $\{C_1, C_2, \dots, C_k\}$  de ses composantes.

## 1.9 Quelques graphes particuliers

On distingue plusieurs types de graphes :

### 1.9.1 Graphe complet

Un graphe  $G = (V, E)$  est dit complet si tous les sommets sont deux à deux adjacents, le graphe complet simple à  $n$  sommets est noté  $K_n$ . [10]  
La figure 1.3 montre le graphe  $K_4$ .

### 1.9.2 Graphe biparti

Un graphe  $G$  est dit biparti si l'ensemble de ses sommets peut-être partitionné en deux sous-ensembles  $V_1$  et  $V_2$  telle que deux sommets de même sous-ensemble ne soient jamais adjacents, et on note parfois  $G = (V_1 \cup V_2, E)$ . [10]

Un sous graphe de graphe biparti est un graphe biparti. La figure 1.4 montre un graphe biparti à 5 sommets et 5 arêtes.

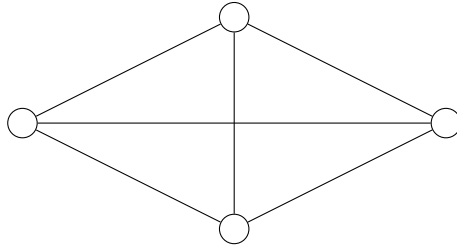


FIGURE 1.3 – Graphe  $K_4$ .

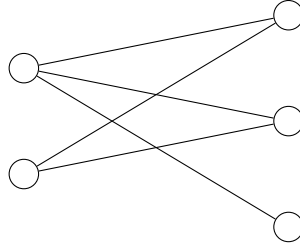


FIGURE 1.4 – Graphe biparti à 5 sommets et 5 arêtes.

### 1.9.3 Graphe biparti complet

Soit  $G = (V_1 \cup V_2; E)$  un graphe biparti. Si pour tout  $v_1 \in V_1$  et pour tout  $v_2 \in V_2$  on a  $v_1$  et  $v_2$  est une arête ( $\forall v_1 \in V_1, \forall v_2 \in V_2, (v_1, v_2) \in E$ ), alors le graphe biparti  $G = (V_1 \cup V_2; E)$  est dit biparti-complet.

Un graphe simple biparti-complet avec  $|V_1| = p, |V_2| = q$  se dénote par  $K_{p,q}$ .<sup>[2]</sup>

La figure 1.5 montre le graphe  $K_{3,3}$ .

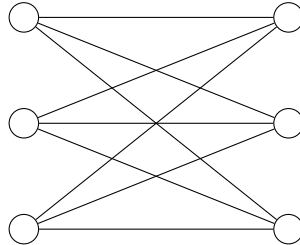


FIGURE 1.5 – Graphe  $K_{3,3}$

### 1.9.4 Graphe biparti équilibré

Un graphe biparti  $G = (V_1 \cup V_2, E)$  est dit équilibré si chaque ensemble de la bipartition est de même taille. Les graphes ayant des cycles de longueur paire, c'est-à-dire les graphes bipartis complets, pour lesquels  $|V_1| = |V_2|$ , sont des exemples de graphes équilibrés.<sup>[17]</sup>

### 1.9.5 Arbre et arborexence

- Un arbre est un graphe non-orienté  $G$  qui vérifie une des conditions équivalentes suivantes :

- $G$  est connexe et acyclique.
- $G$  est sans cycle et possède  $n - 1$  arêtes.
- $G$  est connexe et admet  $n - 1$  arêtes.
- $G$  est sans cycle et si en ajoutant une arête, On crée un seul cycle élémentaire, et si on supprime une arête, on obtient un graphe non connexe.
- Il existe une seule chaîne entre deux sommets quelconques de  $G$ .

- Une arborescence est un arbre comportant un sommet particulier  $r$ , nommé racine de l'arborescence à partir duquel il existe un chemin unique vers tous les autres sommets.[19]

### 1.9.6 Graphe régulier

Un graphe  $G$  est dit  $k$ -régulier si  $\forall x$  sommet de  $G$ , on a  $d_G(v) = k$ . En d'autres termes,  $\delta(G) = \Delta(G) = k$ . La figure 1.6 montre un graphe 3-régulier

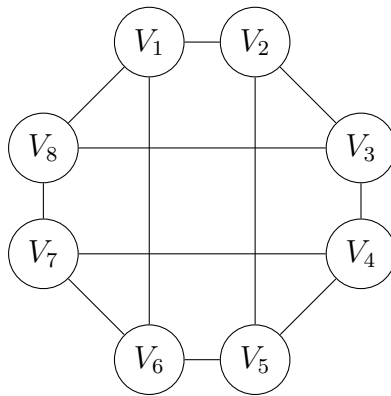


FIGURE 1.6 – Graphe 3-régulier

### 1.9.7 Graphe de Hamming

Soient  $a_1, a_2, \dots, a_n$  des entiers positifs. Le graphe de Hamming  $H a_1, a_2, \dots, a_n$  est le graphe dont les sommets sont les éléments de la somme cartésienne d'ensembles de  $a_i$  éléments et où deux sommets sont adjacents si et seulement si les vecteurs correspondants diffèrent en exactement une seule composante.

### 1.9.8 Graphe planaire

Un graphe  $G$  est dit un graphe planaire si on peut le dessiner sur un plan sans qu'aucune arête ne se croise avec une autre. Soit le graphe  $G$  présenté dans la figure 1.7 :

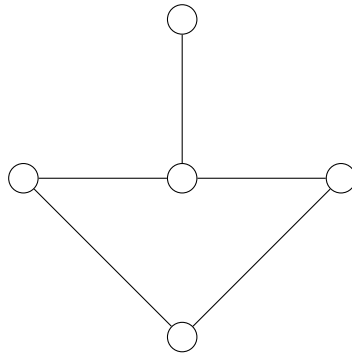


FIGURE 1.7 – Graphe planaire

### 1.9.9 Hypercube

Un hypercube de dimension  $n$  noté  $(Q_n)$  est le graphe de  $2^n$  sommets qui pouvant être considérés comme étant tous les vecteur booléens sur  $(0, 1)^n$ , deux sommets sont adjacents si et seulement si les vecteurs associés à ces sommets diffèrent exactement en une seule composante.

Notons que  $Q_0 = K_1$ ,  $Q_1 = K_2$  et que d'une manière générale.  $Q_n$  peut être défini récursivement en utilisant le produit cartésien par  $Q_{n+1} = Q_n \square K_2$ . [18]

Il est donc clair que  $Q_n (n \geq 1)$  est isomorphe à

$$\underbrace{K_2 \square K_2 \dots \square K_2}_{n \text{ fois}}.$$

Et donc que  $Q_{n+d} = Q_n \square Q_d$ .

La figure 1.8 montre les hypercubes  $Q_0, Q_1, Q_2$  et  $Q_3$ .

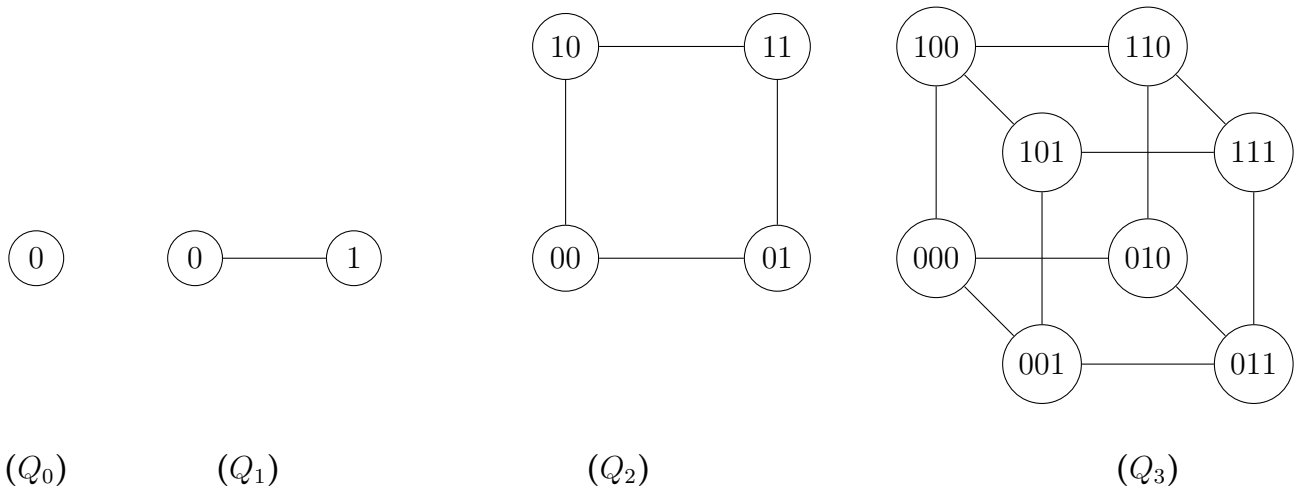


FIGURE 1.8 – les quatre premiers hypercubes

où,

- $V(Q_0) = \{0\}$
- $V(Q_1) = \{0, 1\}$ .
- $V(Q_2) = \{0, 1\} \times \{0, 1\} = \{00, 01, 10, 11\}$ .
- $V(Q_3) = \{00, 01, 10, 11\} \times \{0, 1\} = \{000, 010, 100, 110, 001, 011, 101, 111\}$ .

### Quelques propriétés élémentaires de l'hypercube

L'hypercube de dimension  $n$  est un graphe équilibré,  $n$ -régulier avec  $2^n$  sommets donc  $n * 2^{n-1}$  arêtes.

De plus, pour deux sommets  $x$  et  $y$  qui sont à distance  $k$  dans  $Q_n$ , il existe  $k!$  chaînes géodisiques entre  $x$  et  $y$ .[\[18\]](#)

**Proposition 1.9.1.** *Il existe  $n - 1$  chaînes de longueur inférieure ou égale à  $n$  et une chaîne de longueur inférieure ou égales à  $n + 1$ , deux à deux sommet-disjointes, entre toute paire de sommets distincts de l'hypercube de dimension  $n$ .*[\[18\]](#)

### Note

A noter que dans l'hypercube de dimension  $n$  ( $n \geq 2$ ), toute paire d'arêtes adjacentes de  $Q_n$  appartient à exactement un cycle de longueur 4. Il est à noter aussi que dans un cycle de l'hypercube, le nombre d'arêtes dans une direction donnée  $i$  est toujours pair.

## 1.10 Représentations d'un graphe

Tout graphe peut être représenté par liste d'adjacences et matrice d'adjacences.

### 1.10.1 Liste d'adjacences

La liste d'adjacence est une structure de données utilisée pour représenter un graphe par une liste chaînée des voisins de chaque sommet d'un graphe, qui seront ordonnées dans un tableau de  $n$  éléments ( $n$  ordre du graphe). [3]

La figure 1.9 montre un graphe et sa liste d'adjacence.

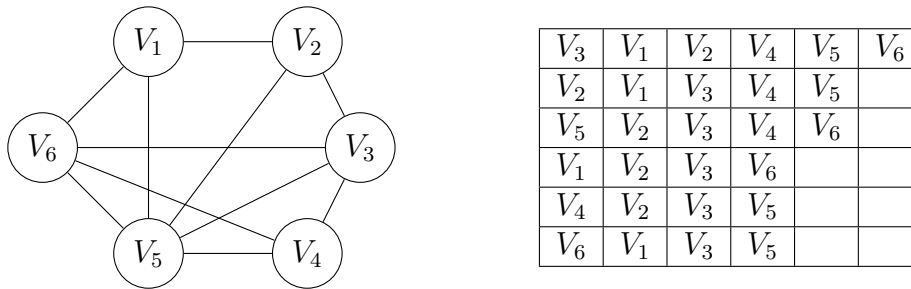


FIGURE 1.9 – Graphe à 6 sommets et 11 arcs et sa liste d'adjacence associée.

### 1.10.2 Matrice d'adjacences

On peut représenter un graphe simple par une matrice d'adjacences. Une matrice ( $n \times m$ ) est un tableau de  $n$  lignes et  $m$  colonnes,  $(i, j)$  désigne l'intersection de la ligne  $i$  et de la colonne  $j$ . Dans une matrice d'adjacences, les lignes et les colonnes représentent les sommets du graphe. Un « 1 » à la position  $(i, j)$  signifie que le sommet  $i$  est adjacent au sommet  $j$ . (et la position  $(i, j)$  représente une arête reliante.) [3]

$$M = \begin{cases} M_{i,j} = 1, & \text{Le sommet } i \text{ est adjacent au sommet } j; \\ M_{i,j} = 0, & \text{Sinon.} \end{cases}$$

La figure 1.10 montre un graphe et sa matrice d'adjacence.



FIGURE 1.10 – Graphe à 4 sommets et 4 arêtes et la matrice d'adjacence associée.

## 1.11 Conclusion

Dans ce chapitre, nous avons rappelé quelques notions de base concernant la théorie de graphe et la complexité algorithmique qui seront utilisées par la suite.



# 2

## Coloration des graphes et coloration des arêtes

### 2.1 Introduction

La coloration de graphes est depuis plusieurs décennies, un domaine très actif de la théorie des graphes du fait de ses nombreuses applications comme l'ordonnancement de tâches, l'allocation de fréquences, l'allocation de registres. En effet, un grand nombre de problèmes peuvent se traduire comme une partition d'un ensemble d'objets suivant certaines contraintes.

Dans ce chapitre nous allons présenter les notions de base de coloration des arêtes et la définition de la parité et forte parité.

### 2.2 Coloration propre

Une coloration propre d'un graphe  $G$  est l'un des outils de la théorie des graphes qui permet de modéliser des problèmes réels, son principe consiste à associer à chaque sommet une couleur, telle que deux sommets adjacents n'aient pas la même couleur.

Le problème d'optimisation classique est de trouver une coloration avec un nombre de couleurs minimum.

### 2.3 Coloration des sommets

On appelle coloration des sommets d'un graphe  $G = (V, E)$ , toute attribution d'une couleur à chaque sommet du graphe. On dit une coloration est **valide** si deux sommets adjacents ont toujours des couleurs différentes. On peut colorier les sommets d'un graphe par l'algorithme de *Walsh and Powell*.

### 2.3.1 Algorithme

---

**Algorithm 4** : Algorithme de Walsh and Powell

---

- (1) Ordonner tout les sommets de  $G$  selon l'ordre des degrés décroissants (une liste).
  - (2) Choisir une nouvelle couleur .
  - (3) Chercher dans la liste le premier sommet non-coloré et le colorier avec la nouvelle couleur.
  - (4) Examiner tour à tour dans l'ordre de la liste tous les sommets non-colorés et colorier chaque sommet non adjacents à un sommet colorié avec la nouvelle couleur et retourner en (2).
- 

### 2.3.2 Nombre chromatique

Le nombre chromatique d'un graphe  $G$  (noté  $\chi(G)$ ) est le nombre minimum  $k$  de couleurs pour donner une coloration propre. C'est à dire le plus petit nombre de couleurs permettant de colorier tous les sommets du graphe sans que deux sommets adjacents soient de la même couleur.

On appelle  $\delta(G)$  le degré maximum des sommets .

Une  $\chi(G)$ -coloration valide est une coloration **optimale**.

Il n'y a pas de formule générale pour déterminer le nombre chromatique d'un graphe. En effet, pour déterminer le nombre chromatique il suffit de déterminer les bornes maximales et minimales pour le nombre chromatique d'un graphe.

#### Encadrement du nombre chromatique

L'algorithme précédent ne conduit pas nécessairement à une coloration minimale c'est-à-dire :  $k \geq \gamma(G)$ . Donc le nombre chromatique généralement inconnue, c'est pour une telle raison que des bornes inférieure  $[\gamma^-(G)]$  et supérieure  $[\gamma^+(G)]$  peuvent être établit pour  $\gamma(G)$  c'est-à-dire :

$$\gamma^-(G) \leq \gamma(G) \leq \gamma^+(G)$$

#### Proposition

Soit  $G = (V, E)$  un graphe simple non orienté avec  $n = |V|$  |  $m = |E|$ .

- $\alpha(G)$  : La taille d'un stable maximale de  $G$ .
- $k$  : La  $k$ -coloration obtenu avec l'algorithme de Powell.
- $d_{max}$  : Le degré maximum dans le graphe  $G$ .
- $d_{min}$  : Le degré minimum dans le graphe  $G$ .
- $\omega(G)$  : La taille d'une clique maximale dans  $G$ , alors le nombre chromatique du graphe  $G$  est compris entre  $\gamma^-(G)$  et  $\gamma^+(G)$ . [21]

où ;

$$\gamma^+(G) = \min\{n + 1 - \alpha(G); k, d_{max} + 1\}$$
$$\gamma^-(G) = \max\left\{\frac{n}{(n - d_{min})}, \omega(G), \frac{n^2}{(n^2 - 2n)}; \left(\frac{n}{\alpha(G)}\right)\right\}$$

## 2.4 Coloration des arêtes

Une coloration des arêtes d'un graphe  $G = (V, E)$  est une affectation de couleurs aux arêtes de  $G$ , telle que deux arêtes incidentes aient des couleurs différentes, alors cette coloration des arêtes est dite une coloration propre des arêtes. La coloration des arêtes de  $G$  revient à la coloration des sommets du graphe représentatif des arêtes de  $G$  (Graphe adjoint), noté  $L(G)$  en utilisant l'algorithme de Walsh and Powell. Considérons le graphe  $G$  de la figure 2.1.

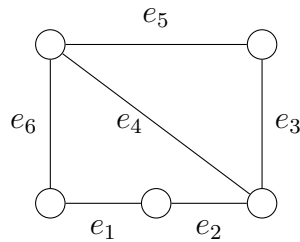


FIGURE 2.1 – Coloration des arêtes

1. Le graphe adjoint de  $G$  est montré dans la figure 2.2.

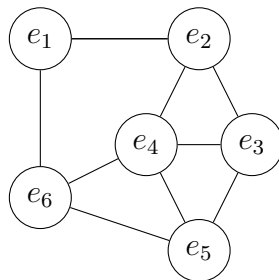


FIGURE 2.2 – Graphe adjoint de  $G$

2. La coloration des sommets de  $L(G)$  est donnée par la tableau ci-dessous

$i$	$e_1$	$e_2$	$e_3$	$e_4$	$e_5$	$e_6$
$d_G(i)$	2	3	3	4	3	3
$\Gamma(C_i)$	6	2	3	1	4	5
$C_i$	R	V	B	R	V	B

Donc le graphe  $G$  est 3-colorable. La 3-coloration de  $L(G)$  est donnée par la figure 2.3(a) et la 3-coloration de  $G$  est donnée par la figure 2.3(b).

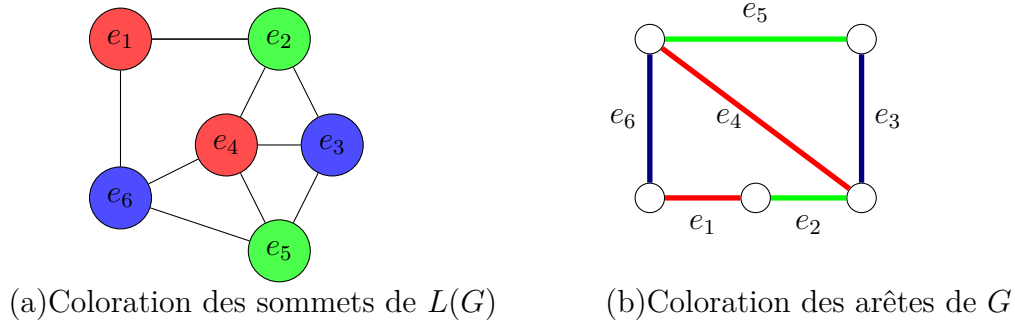


FIGURE 2.3 – 3-coloration de  $L(G)$

### 2.4.1 Indice chromatique

L'indice chromatique est le nombre de couleurs à affecter aux arêtes d'un graphe  $G$  telle que deux arêtes incidentes n'aient pas la même couleur.

Dans le cas où le nombre de couleur à attribuer aux arêtes de  $G$  est minimale, on parlera d'indice chromatique fort, noté  $\chi'(G)$ . L'indice chromatique et l'indice chromatique fort de l'hypercube de dimension 2 sont donnés respectivement par la figure 2.4.

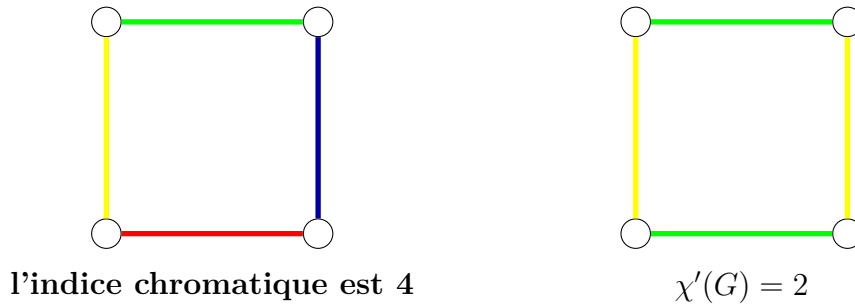


FIGURE 2.4 – Indice Chromatique de l'hypercube  $Q_2$ .

### 2.4.2 Encadrement de l'indice chromatique

Il n'y a pas de formule générale pour déterminer l'indice chromatique d'un graphe. En effet, plusieurs théorèmes permettant l'encadrement de l'indice chromatique ont été donnés, on peut citer le théorème de Vizing, le théorème de Koing et le théorème de Gupta.

**Théorème 2.4.1.** *Le théorème de Vizing permet de déterminer une coloration des arêtes de  $G$  comprise entre  $\Delta(G)$  et  $\Delta(G) + 1$  couleurs. Son algorithme se trompe donc éventuellement d'une unité sur l'indice chromatique, ce qui n'est pas énorme, on aura :[?]*

$$\Delta(G) \leq \chi'(G) \leq \Delta(G) + 1$$

. Ce théorème répartit les graphes finis en deux classes selon leur indice chromatique, comme le montre le tableau suivant :

Classe 1	Classe 2
$\chi'(G) = \Delta(G)$	$\chi'(G) = \Delta(G) + 1$

**Théorème 2.4.2.** (König)

König en 1916, a prouvé que l'indice chromatique fort d'un graphe biparti  $G$  est égal au degré maximum de  $G$ . c'est à dire  $\chi'(G) = \Delta(G)$ . [25]

L'indice chromatique fort de graphe  $K_{1,3}$  est montré dans la figure 2.5.

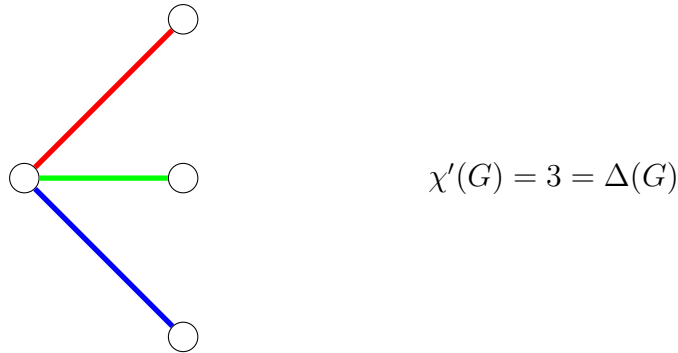


FIGURE 2.5 – Graphe  $K_{1,3}$

**Théorème 2.4.3.** (Gupta)

Gupta regroupe les deux résultats précédents et affirme que pour tout graphe simple  $G$  on a :  $\chi'(G) = \Delta(G)$  ou  $\chi'(G) = \Delta(G) + 1$ . [1]

**2.4.3 Propriétés**

Si  $\chi(G)$  est le degré maximum de  $G$  alors :

- $\chi'(G) = 1$  si et seulement si  $G$  est un couplage.
- $\chi'(G) \geq \Delta(G)$ .
- $\chi'(G) = \Delta(G)$  dès que  $G$  est un graphe simple planaire tel que  $\Delta(G) \geq 7$ .

On peut vérifier facilement les propriétés précédentes dans les graphes suivants :

- L'indice chromatique fort de l'hypercube de dimension 2 et de graphe triangulé données par la figure 2.6 sont respectivement  $\chi'(G) = \Delta(G) = 2$  et  $\chi' = 3 \geq \Delta = 2$ .

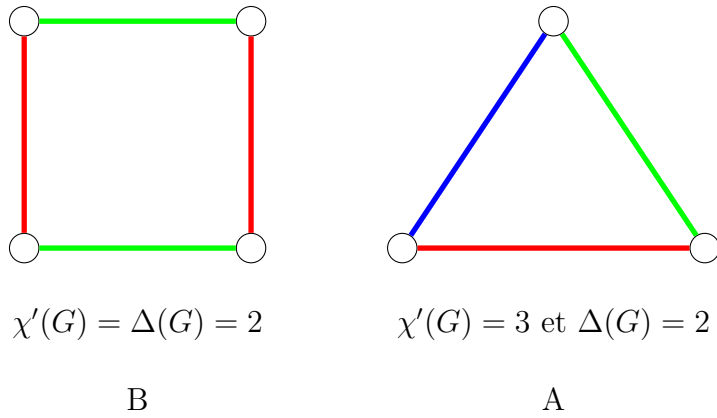


FIGURE 2.6 – Indice chromatique fort.

- Soit la figure 2.7

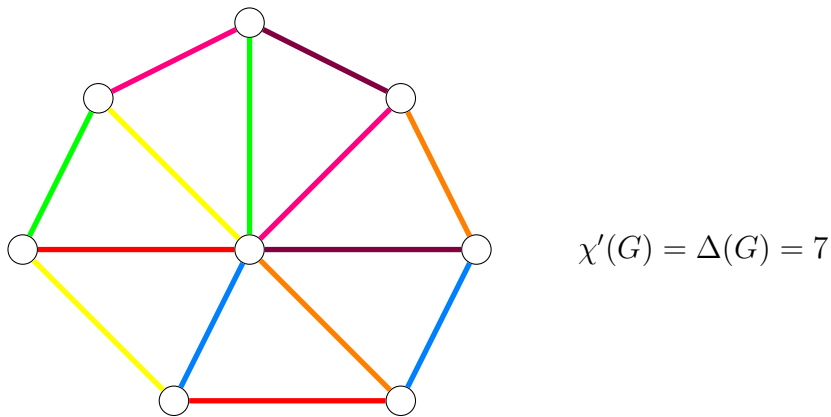


FIGURE 2.7 – Graphe planaire de degré 7.

**Lemme 1.** Soit  $G$  un graphe simple,  $v$  un sommet de  $G$ ,  $e$  une arête de  $G$  incidente à  $v$ , et  $k$  un entier avec  $k \geq \Delta$ . Supposons que  $\chi'(G \setminus e) = k$  telle que tout voisin de  $v$  a au moins une couleur disponible. Alors  $\chi'(G) = k$ . [1]

On distingue tout types d'indice chromatique fort :

- Parité de l'indice chromatique, notée  $P(G)$  est l'indice chromatique  $\chi'(G)$  tel que pour toute chaîne (path)  $P$  de  $G$ ,  $\exists k \in \{1, 2, 3, \dots, \chi'(G)\}$  pour lequel un nombre impair d'arêtes de  $P$  sont marquées par  $k$ .

Un résultat concernant les arbres binaires à états donnée par Ivan Havel.

**Lemme 2.** Un arbre  $T$  est isomorphe à un sous graphe de l'hypercube de dimension  $n$  si et seulement si  $P(T) \leq n$ . [?]

La figure 2.8 montre le graphe  $G$  et son indice chromatique  $P(G) = 4$ .

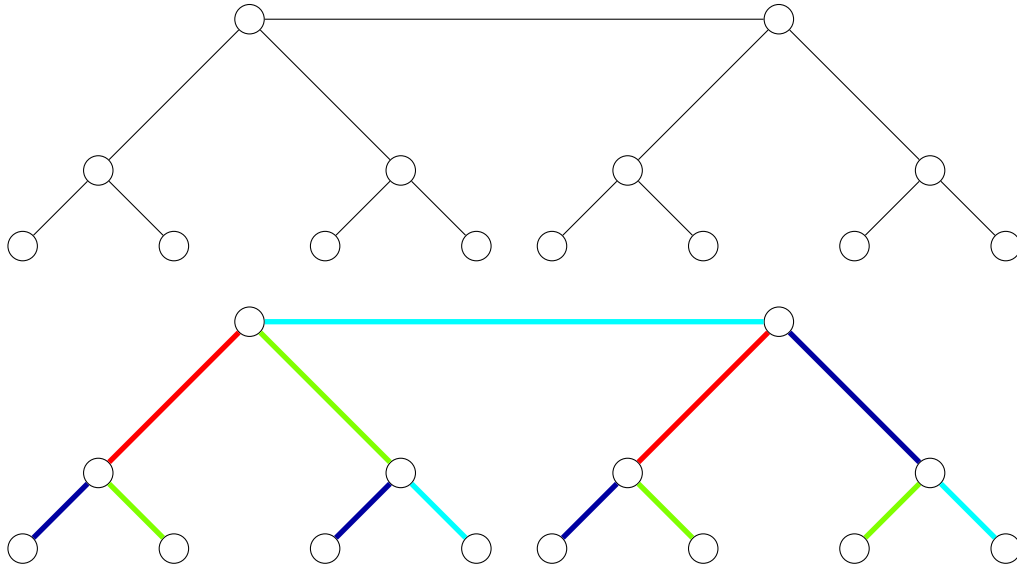


FIGURE 2.8 – Arbre  $T$  et son indice chromatique  $P(G)$ .

- Forte parité de l'indice chromatique , notée  $\hat{P}(G)$  est l'indice chromatique  $\chi'(G)$  tel que pour tout cycle  $C$  de  $G$ ,  $\forall k \in \{1, 2, 3, \dots, \chi'(G)\}$ ,  $k$  est observé un nombre pair de fois. La figure 2.9 montre la forte parité de l'indice chromatique d'un hypercube  $Q_3$ , qui est  $\hat{P}(Q_3)$ .

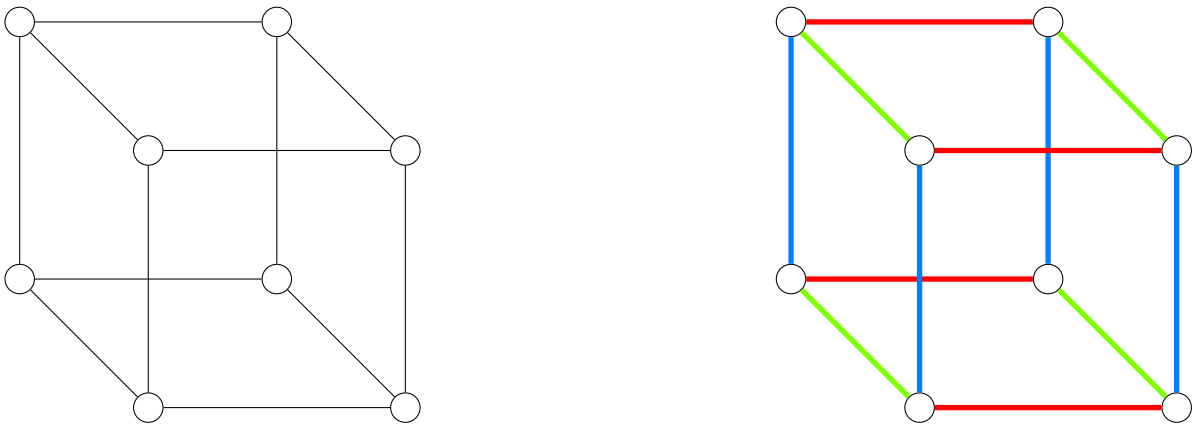


FIGURE 2.9 –  $\hat{P}(Q_3) = 3$ .

## 2.5 Parité et forte parité de quelques arbres binaires

### 1. Arbre binaire complet $D_n$ .

On définit l'arbre  $D_n$  inductivement comme suit :

$D_1$  est le graphe biparti  $K_{1,2}$ . Pour  $n \geq 2$ ,  $D_n$  est obtenu à partir de  $D_{n-1}$  tel que chaque sommet pendant de  $D_{n-1}$  est relié à deux nouveaux sommets. L'unique sommet de degré 2 de  $D_n$  est appelé racine de  $D_n$ .

La figure 2.10 montre les arbres  $D_1$ ,  $D_2$  et  $D_3$ .

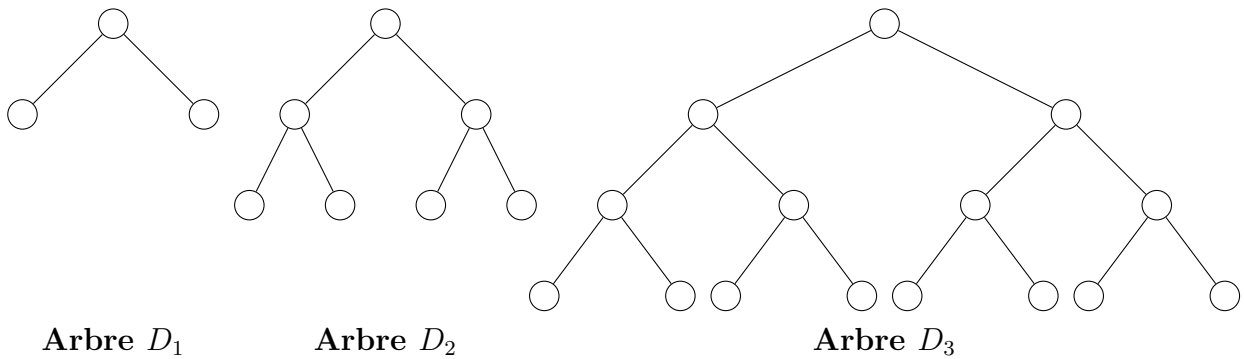


FIGURE 2.10 – Arbres  $D_1$ ,  $D_2$ ,  $D_3$ .

Havel a montré que  $P(D_1) = 2$  et  $\forall n \geq 2$ ,  $P(D_n) = n + 2$  c'est-à-dire les arêtes de  $D_1$  sont coloriées par les entiers de l'ensemble  $\{1, 2\}$  et les arêtes de  $D_n$  sont coloriées par les entiers de l'ensemble  $\{1, 2, \dots, n, n + 1, n + 2\}$  tel que pour toute chaîne  $P$  de  $D_n$   $\exists k \in \{1, 2, \dots, n, n + 1, n + 2\}$  pour lequel un nombre impair d'arêtes de  $P$  sont marquées par  $k$ . D'après Vizing[19]  $P(D_1) = \hat{P}(D_1) = 2$  et  $P(D_n) = \hat{P}(D_n) = n + 2$   $\forall n \geq 2$  car  $D_n$  est biparti.

La figure 2.11 montre  $P(D_1)$  et  $P(D_2)$ .

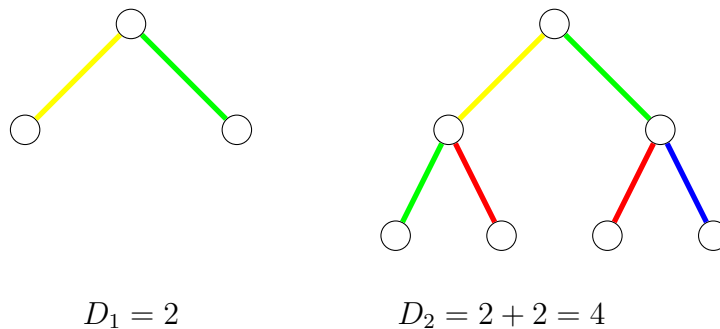


FIGURE 2.11 –  $\hat{P}(D_1)$  et  $\hat{P}(D_2)$ .

Comme  $P(D_n) \leq n + 2$  alors  $D_n$  est isomorphe à un sous graphe de  $Q_{n+2}$ . Voir la figure 2.12 qui montre le plongement de l'arbre  $D_2$  dans l'hypercube  $Q_4$ .



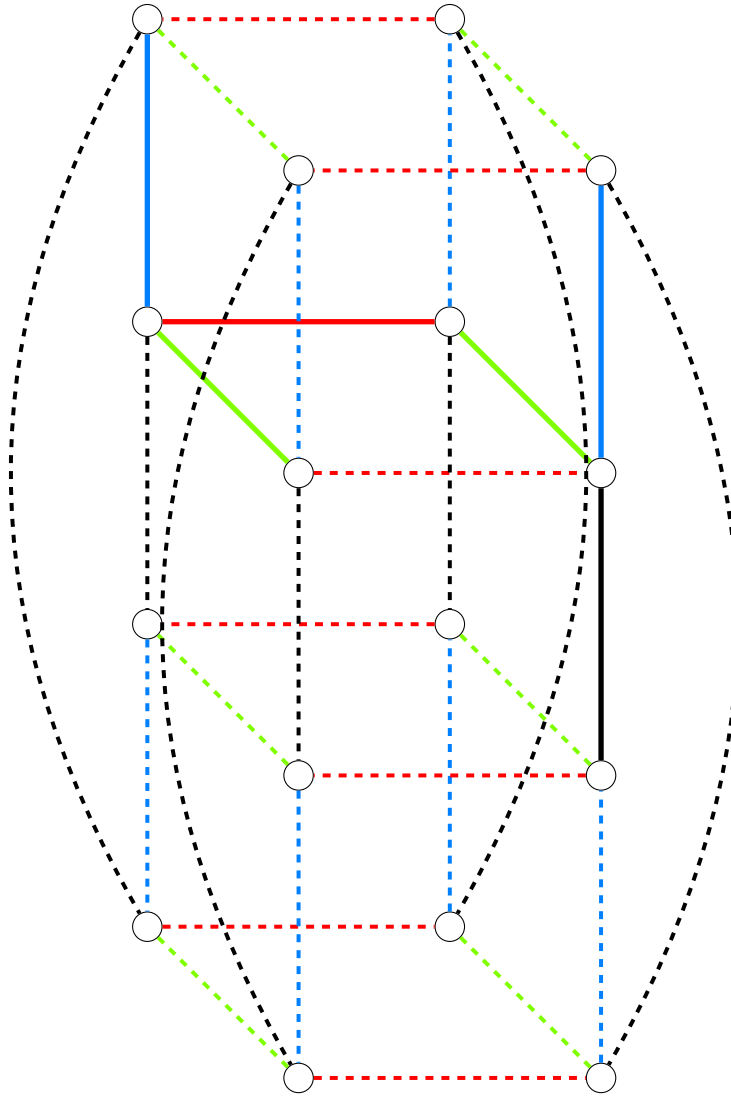


FIGURE 2.12 – Plongement de l'arbre  $D_2$  dans  $Q_4$ .

2. **Arbre binaire  $B_n$**

Pour toute  $n \geq 2$  l'arbre  $B_n$  est obtenu à partir de l'arbre  $D_{n-1}$  et un sommet  $u$  tel que  $u$  est relié par une arête à la racine de  $D_{n-1}$ .

$B_2$  et  $B_3$  sont montrés dans la figure 2.13.

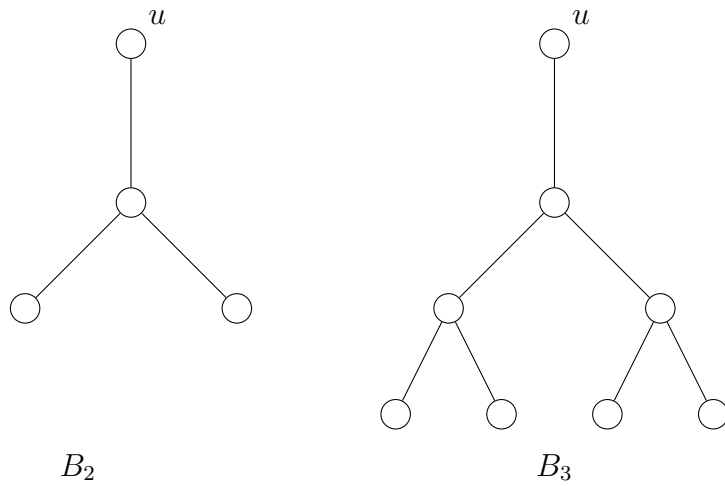


FIGURE 2.13 – Arbre  $B_2$  et arbre  $B_3$ .

Havel et Liebl ont montré que  $n \geq 2$ ,  $P(B_n) = n + 1$ . D'après le Lemme 2  $B_n$  est un sous graphe de  $Q_{n+1}$ .  $P(B_2)$  et  $P(B_3)$  sont montrés dans la figure 2.14.

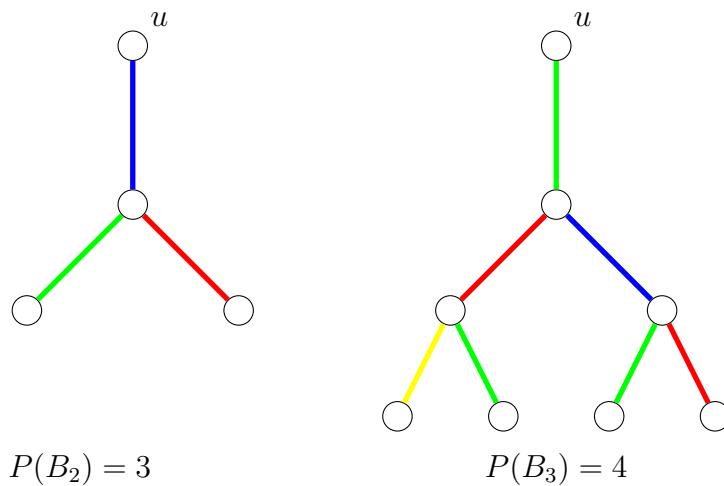


FIGURE 2.14 –  $P(B_2)$  et  $P(B_3)$ .

La figure 2.15 montre le plongement de l'arbre  $B_2$  dans son hypercube optimal  $Q_3$ .

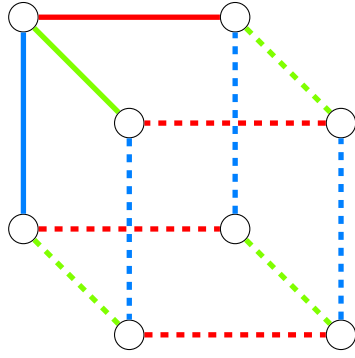


FIGURE 2.15 – Plongement de l'arbre  $B_2$  dans  $Q_3$ .

# 3

## *Parité et forte parité de l'Indice Chromatique de deux nouvelles classes d'arbres et application*

### **3.1 Introduction**

*Dans ce chapitre nous allons caractériser deux nouvelles classes d'arbre pour lesquelles la parité de l'indice chromatique  $P$  et la forte parité de l'indice chromatique  $\hat{P}$  sont déterminées. Puis nous allons montrer que chacune des deux classes est plongéable dans son hypercube optimal (hypercube de dimension minimale). A la fin, nous allons présenter l'implémentation de l'algorithme de Prim en utilisant le code block  $c^{++}$ .*

### **3.2 Caractérisation de deux nouvelles classes d'arbres**

*Dans notre étude nous allons caractériser des nouvelles classes d'arbre pour lesquelles nous déterminerons  $P(G)$  et  $\hat{P}(G)$ .*

#### **3.2.1 Arbre $\hat{B}_n$**

*On définit l'arbre  $\hat{B}_n$  inductivement comme suit :*

*$\forall n \geq 2$ , l'arbre  $\hat{B}_n$  est obtenu à partir de  $B_n$  en subdivisant deux fois l'arête verticale (arête incidente à  $u$ ) de  $B_n$  et une seule fois chaque arête de  $D_{n-1}$ .*

*Il est clair que  $\hat{B}_n$  est équilibré et possède  $2^{n-1} - 1$  sommets de degré 3,  $2^n$  sommets de degré 2 et  $2^{n-1} + 1$  sommets pendants, donc au total  $\hat{B}_n$  possède  $2^{n+1}$  sommets. Les*

arbres  $\widehat{B}_2$  et  $\widehat{B}_3$  sont donnés par la figure 3.1.

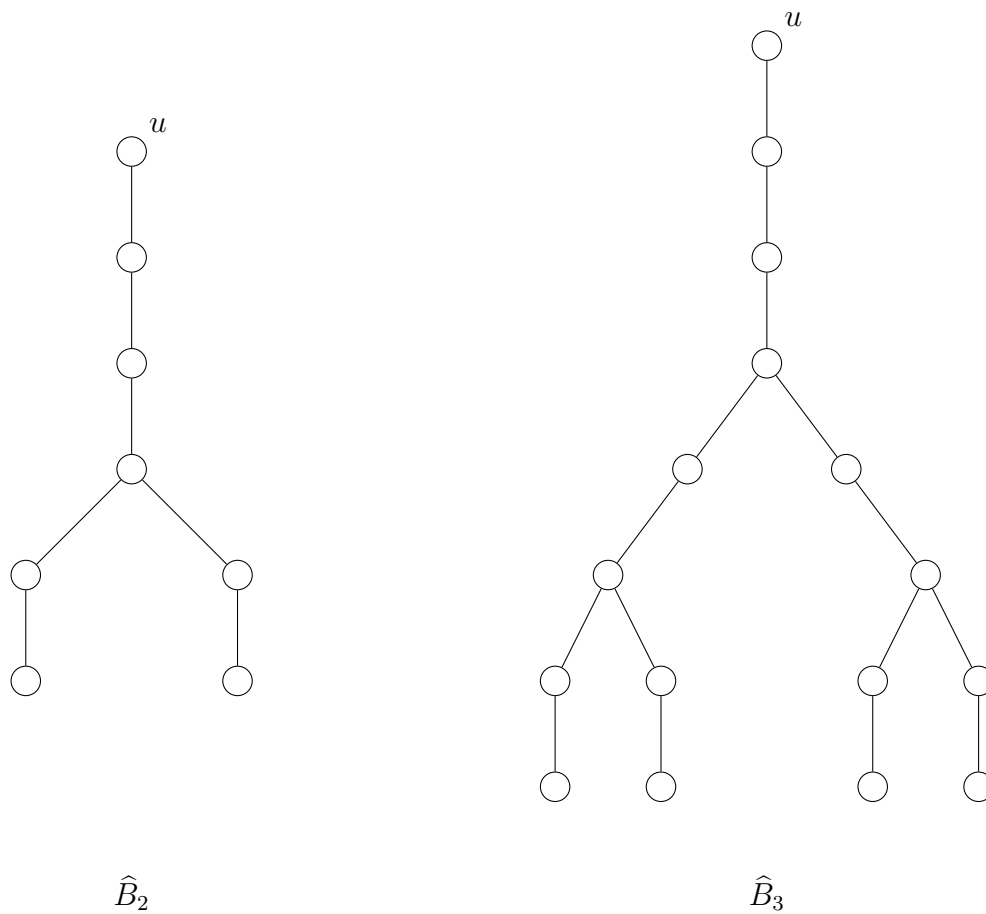


FIGURE 3.1 – Arbres  $\widehat{B}_2$  et  $\widehat{B}_3$ .

**Théorème 3.2.1.** *Pour tout  $n \geq 2$ ,  $P(\widehat{B}_n) = \widehat{P}(\widehat{B}_n) = n + 1$ .*

***Preuve***

*Nous allons utiliser la récurrence sur  $n$ . Pour  $n = 2$ , la 3-colorations des arêtes de  $\widehat{B}_2$  tel que pour toute chaîne  $P$  de  $\widehat{B}_2$ ,  $\exists k \in \{1, 2, 3\}$  pour lequel un nombre impair d'arêtes de  $P$  sont marquées par  $k$  est donnée par la figure 3.2*

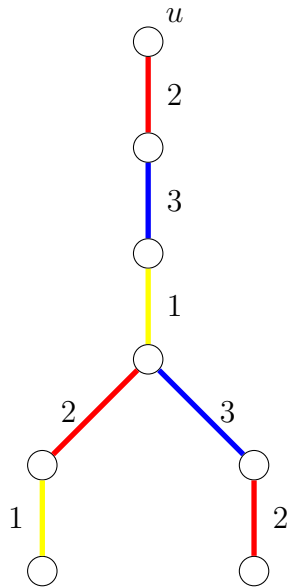


FIGURE 3.2 – Coloration des arêtes de l'arbre  $\widehat{B}_2$ ,  $\alpha(\widehat{B}_2) = 3$ .

En permutant les entiers de l'ensemble  $\{1, 2, 3\}$  deux à deux, nous obtenons deux nouvelles coloration des arêtes de  $\widehat{B}_2$ , voir la figure 3.3.

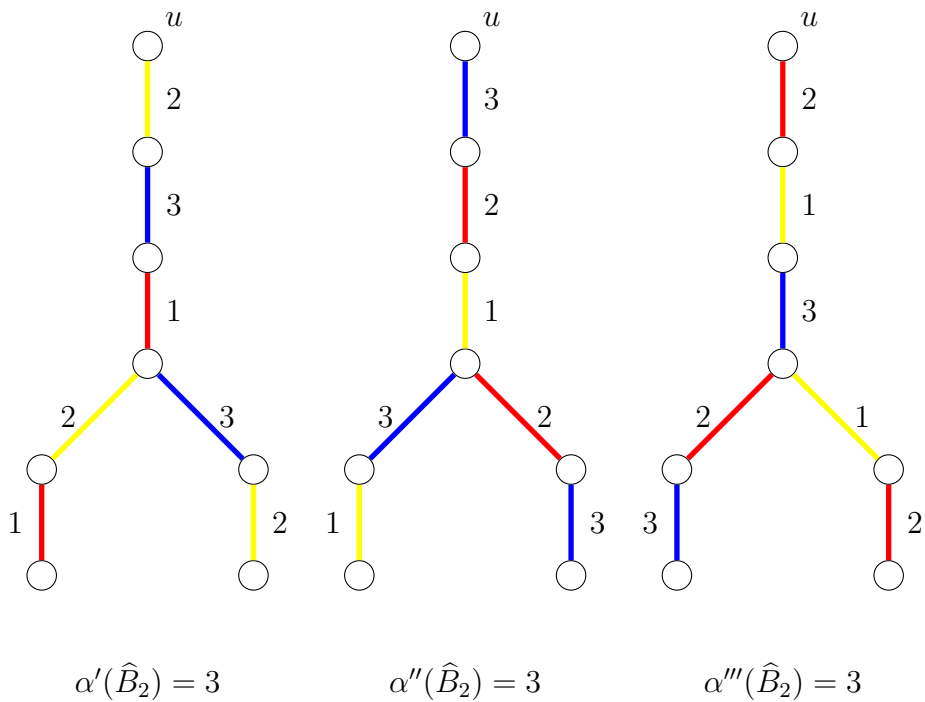


FIGURE 3.3 – Les permutations possible de l'arbre  $\widehat{B}_2$ .

Avec,  $\alpha'(\widehat{B}_2) = 3$  est l'arbre obtenu à partir de l'arbre  $\widehat{B}_2$  en permutant entre 1 et 2,  $\alpha'''(\widehat{B}_2) = 3$  est l'arbre obtenu à partir de l'arbre  $\widehat{B}_2$  en permutant entre 2 et 3 et

$\alpha''(\widehat{B}_2) = 3$  est l'arbre obtenu à partir de l'arbre  $\widehat{B}_2$  en permutant entre 1 et 3.

La coloration des arêtes de  $\widehat{B}_n$  vérifiant  $P(\widehat{B}_n) = n + 1$  est donnée par la figure 3.4.

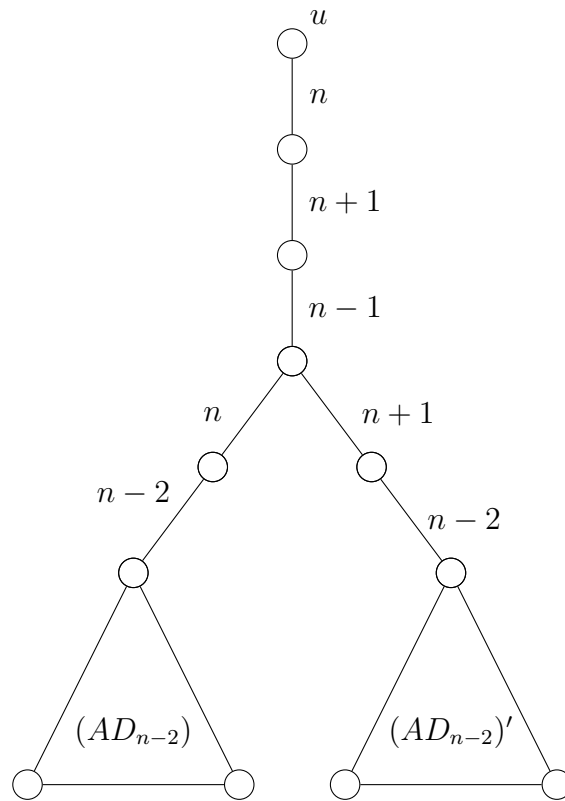


FIGURE 3.4 – Coloration des arêtes des arbres  $\widehat{B}_n$ .

avec  $(AD_n)$  est l'arbre  $D_n$  obtenue à partir de l'arbre  $D_n$  en subdivisant ses arêtes une seule fois.

Tel que la coloration des arêtes de  $AD_{n-2}$  est donnée par  $\alpha(AD_{n-2})$  dans  $\widehat{B}_{n-1}$  et la coloration de l'arbre  $(AD_{n-2})'$  est  $\alpha'(AD_{n-2})$  dans  $\widehat{B}_{n-1}$ .

La coloration de  $\widehat{B}_3$  et  $\widehat{B}_4$  sont présentés par la figure 3.5 :

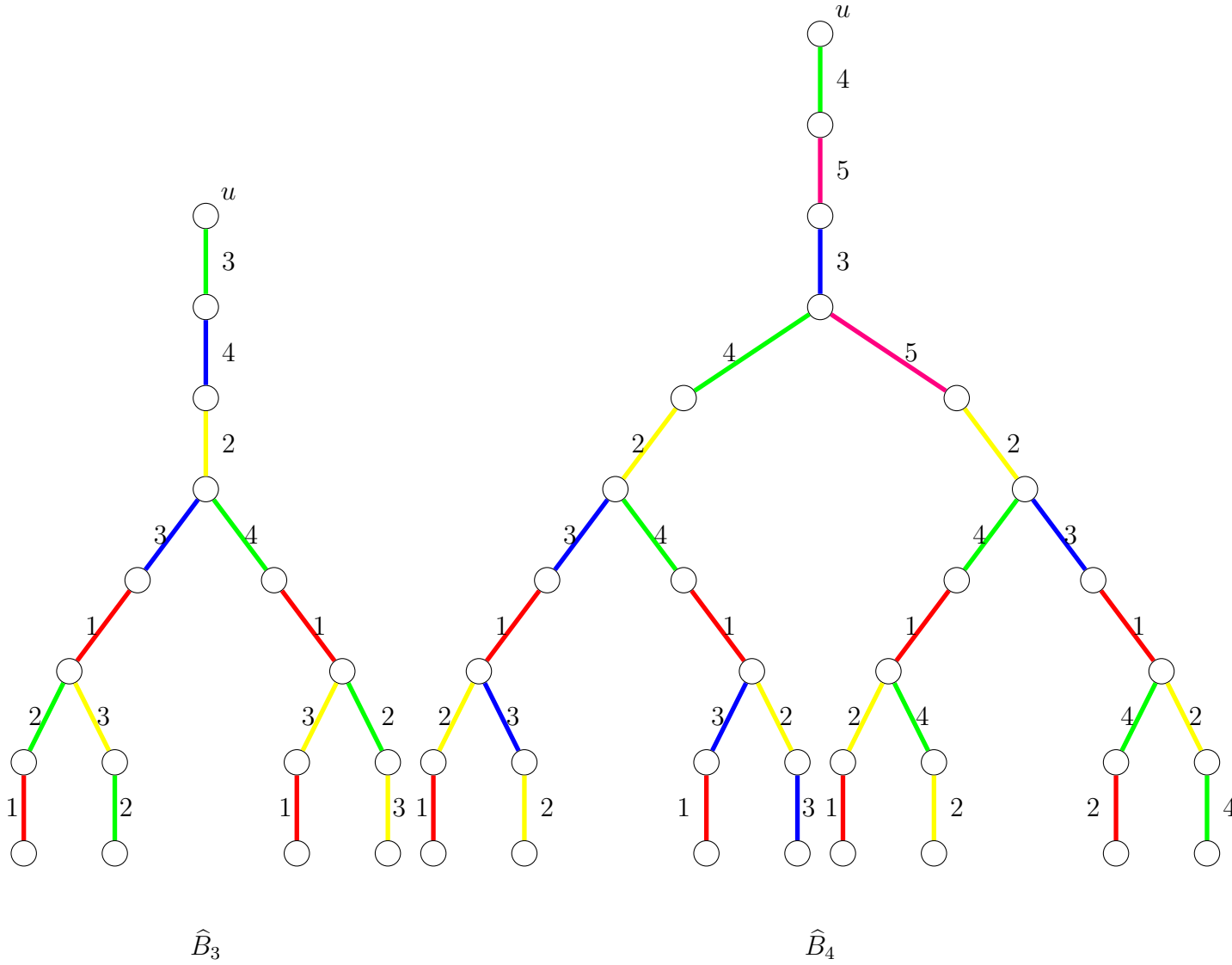


FIGURE 3.5 – Coloration des arêtes des arbres  $\hat{B}_3$  et  $\hat{B}_4$ .

Il est clair que :

- Si une chaîne de  $\hat{B}_n$  utilise une fois  $n + 1$ , alors  $P(\hat{B}_n) = n + 1$  car  $n + 1$  est observé un nombre impair de fois (une fois).
- Si la chaîne  $P$  de  $\hat{B}_n$  n'utilise pas la couleur  $n + 1$ , alors  $P \in \hat{B}_{n-1}$ . Comme  $P(\hat{B}_{n-1}) = n$  (hypothèse de récurrence), alors  $\exists k \in \{1, \dots, n\}$  tel que  $k$  est observé un nombre impair de fois.
- Si  $P$  utilise  $n + 1$  deux fois, alors si on remplace la chaîne de  $\hat{B}_n$  marquée par  $n + 1, n - 1, n + 1$  par une arête marquée  $n - 1$ , on obtient une chaîne  $P' \in \hat{B}_{n-1}$ , donc  $\exists k \in \{1, \dots, n\}$  pour laquelle un nombre impair d'arêtes de  $P'$  sont marquées par  $k$ . Comme  $\hat{B}_n$  est biparti, alors d'après Vizing on a  $P(\hat{B}_n) = P(B_n) = n + 1$ .

**Corollaire 3.1.** Pour toute  $n \geq 2$  l'arbre  $\hat{B}_n$  est plongeable dans son hypercube optimal de dimension  $n + 1$ ,  $(Q_{n+1})$ .



### Preuve

Comme  $P(\widehat{B}_n) = n + 1$ , alors d'après Bunde [11]  $\widehat{B}_n$  est plogeable dans  $Q_{n+1}$ . De plus le nombre de sommets de  $\widehat{B}_n$  est  $2^{n+1}$  ; alors  $\widehat{B}_n$  ne peut pas être plogeable dans  $Q_n$ , car  $2^{n+1} > 2^n$  donc le plus petit hypercube qui contient  $\widehat{B}_n$  est  $Q_{n+1}$ .

### 3.2.2 Arbre $T_n$

On définit l'arbre  $T_n$  inductivement comme suit :  
Pour  $n = 1$ , l'arbre  $T_1$  est donné par la figure 3.6

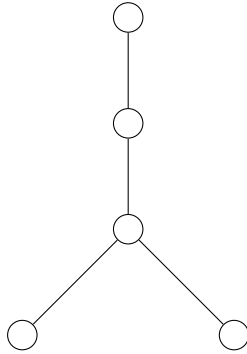


FIGURE 3.6 – Arbre  $T_1$ .

Pour  $n \geq 2$ , l'arbre  $T_n$  est obtenu à partir de deux copies disjointes  $T'_{n-1}$  et  $T''_{n-1}$  tel que un sommet de degré  $n + 1$  relié à un seul sommet pendant dans  $T'_{n-1}$  est relié par une arête à son analogue dans  $T''_{n-1}$ .  
les arbres  $T_2$  et  $T_3$  sont montrés respectivement par la figure 3.7 et la figure 3.8.

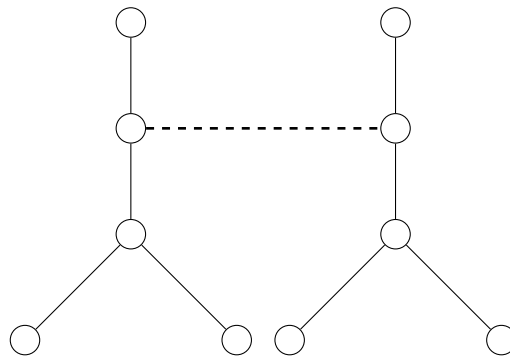


FIGURE 3.7 – Arbre  $T_2$ .

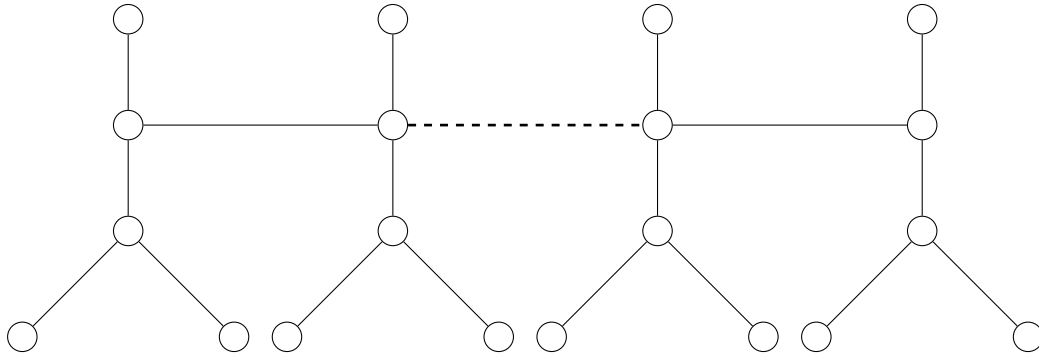


FIGURE 3.8 – Arbre  $T_3$ .

**Théorème 3.2.2.** *Pour tout  $n \geq 1$  :  $P(T_n) = \hat{P}(T_n) = n + 2$ .*

**Preuve**

*Nous allons utiliser la récurrence sur  $n$ .*

*Il est clair que  $P(T_1) = 3$  et  $P(T_2) = 4$  comme le montre le graphe de la figure 3.9*

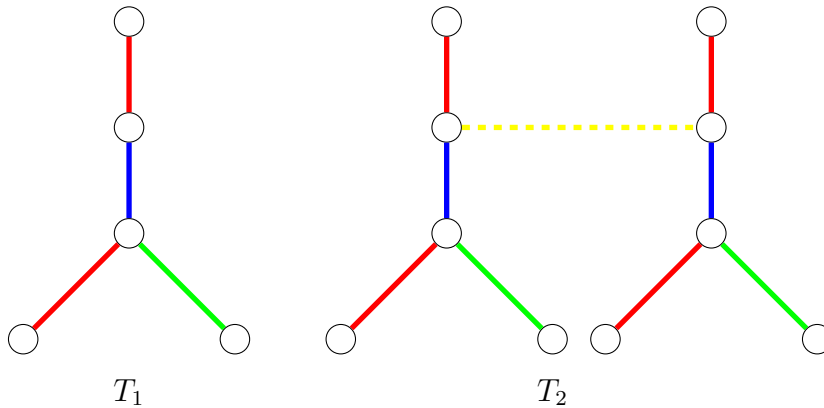


FIGURE 3.9 – Coloration des arêtes des arbres  $T_1$  et  $T_2$ .

*La propriété est vérifiée pour  $n = 1$  et  $n = 2$ . Supposons que  $P(T_{n-1}) = n-1+2 = n+1$ . Comme  $T_n$  est obtenu à partir de deux copies disjointes  $T'_{n-1}$  et  $T''_{n-1}$  tel que un sommet de degré  $n + 1$  relié à un seul sommet pendant dans  $T'_{n-1}$  est relié par une arête à son analogue dans  $T''_{n-1}$ . Alors si on marque l'arête reliant  $T'_{n-1}$  et  $T''_{n-1}$  par  $n + 2$  on obtient  $P(T_n) = n + 2$ . Comme  $T_n$  est biparti alors  $P(T_n) = \hat{P}(T_n) = n + 2$ .*

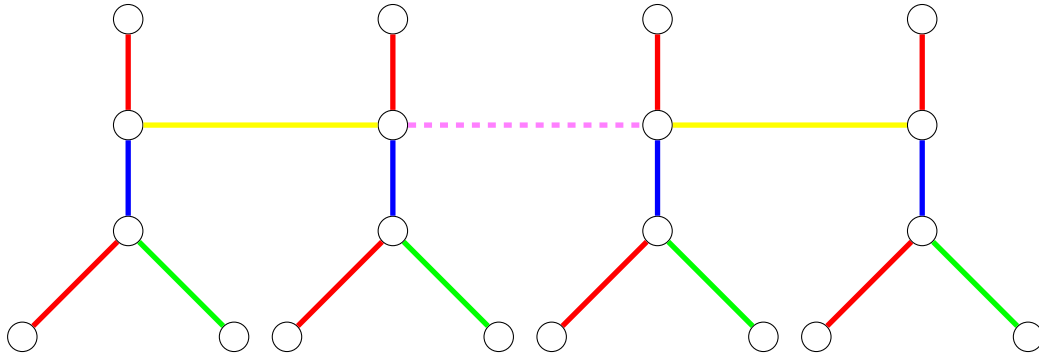


FIGURE 3.10 – Coloration des arêtes de l'arbre  $T_3$ .

Pour le cas général, la coloration de l'arbre  $T_n$  est montré dans la figure 3.11

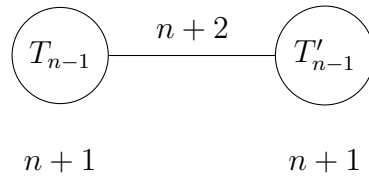


FIGURE 3.11 – Coloration des arêtes de l'arbre  $T_n$ .

**Corollaire 3.2.** Pour toute  $n \geq 1$ , l'arbre  $T_n$  est plongéable dans son hypercube optimal de dimension  $n + 2$ ,  $(Q_{n+2})$ .

**Preuve**

Comme  $P(T_n) = \hat{P}(T_n) = n + 2$  et d'après Koing alors  $T_n$  est plongéable dans  $Q_{n+2}$ . De plus nous avons  $|V(T_n)| = 5 \times 2^{n-1} = (2^2 + 1) \cdot 2^{n-1} = 2^{n+1} + 2^{n-1}$  sommets, alors  $T_n$  ne peut pas être plongéable dans  $Q_{n+1}$  car le nombre de sommets de  $T_n$  dépasse le nombre de sommets de  $Q_{n+1}$ , donc le plus petit hypercube contenant  $T_n$  est  $Q_{n+2}$ .

### 3.3 Application dans Code Blocks

Pour montrer que  $P(T) = k$ , il faut montrer que  $T$  est isomorphe à un sous graphe de  $Q_k$ , on peut étudier un cas pratique pour lequel nous déterminons un arbre de poids minimum en utilisant l'algorithme de Prim.

**Langage utilisé**

**C** est un langage de programmation impératif et généraliste. Inventé au début des années 1970 pour réécrire UNIX, C est devenu un des langages les plus utilisés. De nombreux langages plus modernes comme C++, C#, Java et PHP reprennent des aspects de C.

Le C est un langage de programmation de bas niveau très populaire, créé dans les années 1970 par D.Ritchie et B.W.Kernighan. Il est portable, libre, faiblement typé (peu de types de variables différents : son fonctionnement est donc proche de l'ordinateur (gain

en rapidité), mais un brin plus difficile à manipuler pour le programmeur). Le C n'est sans doute pas le langage le plus facile à apprendre (notamment à cause de l'adoption du concept parfois un peu obscure des pointeurs), ni le plus récent, mais ses qualités font de lui un langage incontournable en matière de programmation.

## Technologies utilisées

**Dev-C++** est un environnement de développement intégré (IDE) permettant de programmer en C et en C++. Développé avec Borland Delphi 6, Dev-C++ était disponible uniquement sous Microsoft Windows. Longtemps à l'abandon, le projet a été repris par un autre développeur en 2011 et est régulièrement mis à jour.

Cet IDE complet comprend entre autres un « répertoire de classes », servant à localiser facilement les fonctions, classes et membres du code source, un « répertoire de fonctions incluses », fonctionnant comme le répertoire de classes mais pour chercher dans les fichiers inclus (header), et un débogueur qui permet de surveiller l'état des variables pendant l'exécution du programme. Il souffre en revanche de l'absence d'un éditeur de ressources, ce qui rend la conception d'applications délicate si on ne fait pas appel à un outil externe.

**Aspect pratique** Utiliser Code : :Blocks sous Windows.

Lorsque on lance Code : :Blocks, nous voyons apparaitre l'écran dans la figure 3.12.

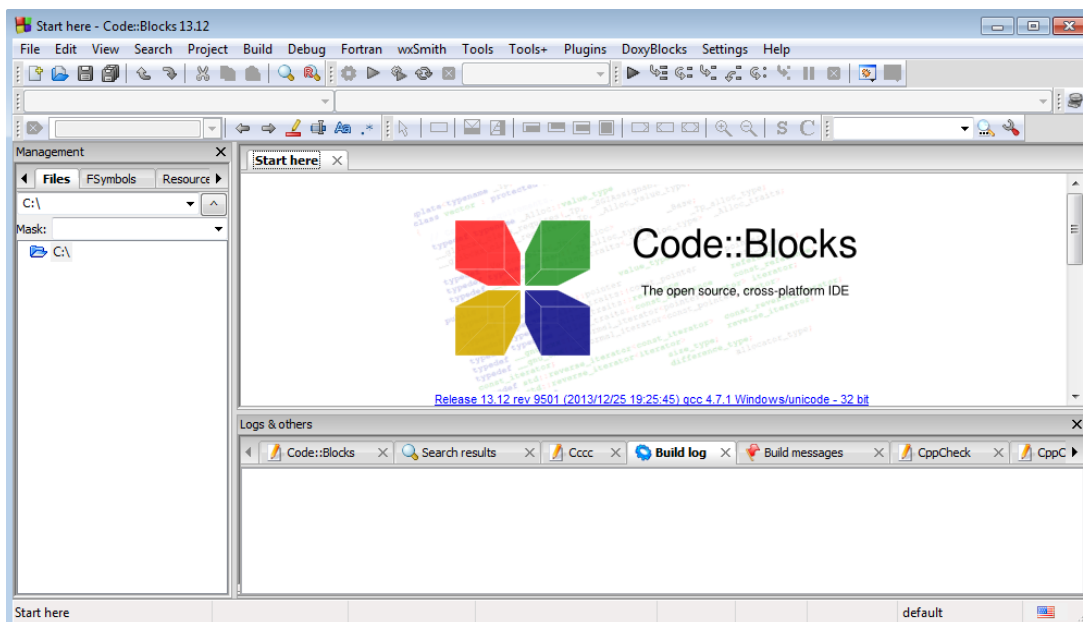


FIGURE 3.12 – l'interface de code blocks.

Pour créer un nouveau projet, il faut choisir dans le menu File puis New puis Project.

## Application de notre implimentation à un problème de communication

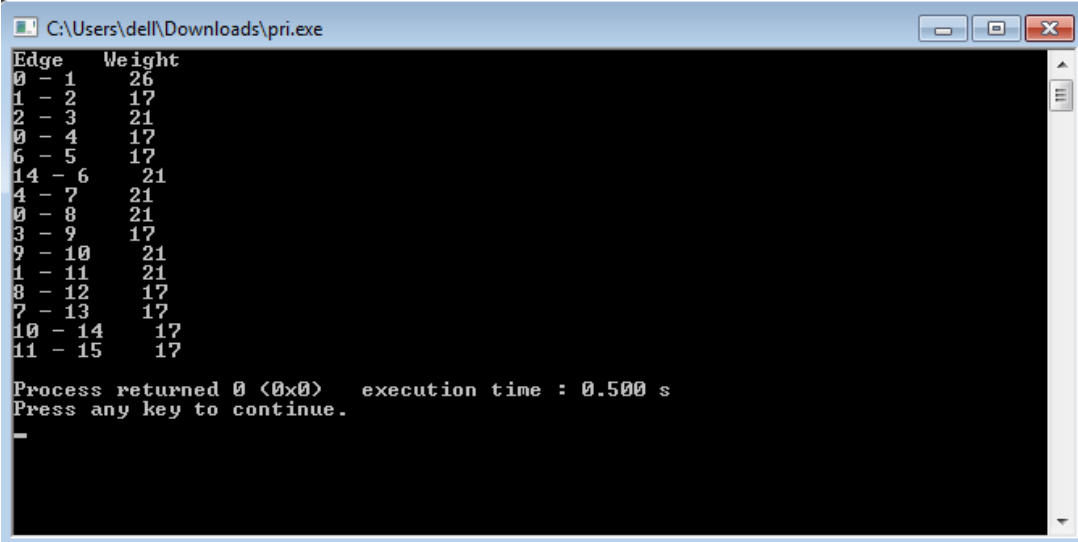
La propagation de nouvelles maladies contagieuses constituait l'un des 10 risques majeurs auxquels le monde se trouvait exposée selon une évaluation coordonnée par le Centre de Crise Mondiale en 2018. La crise sanitaire que nous traversons donne cruellement raison aux experts. Ceux-ci avaient attribué une haute probabilité à un événement de cette nature associé à un impact humain, sociétal, économique et environnemental considérable. Pour se faire fassent chaque pays doit agir immédiatement pour se préparer, répondre et se relever pour éradiquer cette épidémie, diverses organisations internationales de santé sont en compétition pour trouver un médicament contre cette maladie. Afin de créer ce vaccin, chaque organisation doit confidentialiser leurs données de telle sort d'assurer que l'information n'est accessible qu'à ceux dont l'accès est autorisé.

Supposons que la topologie de réseau associé à notre problématique est donnée par un hypercube de dimension 4 ( $Q_4$ ), tel que  $v_1v_2 \in E(G)$  si une communication entre les personnes est possible (de  $v_i$  à  $v_j$  et vice versa). Il existe une probabilité  $P_{ij}$  d'interception d'un message transmis entre  $v_i$  et  $v_j$ . Donc le problème revient à trouver un arbre couvrant  $T$  plongeable dans  $Q_4$  qui minimise la probabilité  $P^* = 1 - \prod_{i,j \in T} (1 - P_{ij})$ . Les probabilités  $P_{ij}$  sont données par le tableau si après :

$S$	$S_1$	$S_2$	$S_3$	$S_4$	$S_5$	$S_6$	$S_7$	$S_8$	$S_9$	$S_{10}$	$S_{11}$	$S_{12}$	$S_{13}$	$S_{14}$	$S_{15}$	$S_{16}$
$S_1$	0	0.26	0	0.36	0.17	0	0	0	0.21	0	0	0	0	0	0	0
$S_2$	0.26	0	0.17	0	0	0.36	0	0	0	0	0	0.21	0	0	0	0
$S_3$	0	0.17	0	0.21	0	0	0.26	0	0	0	0.36	0	0	0	0	0
$S_4$	0.36	0	0.21	0	0	0	0	0.26	0	0.17	0	0	0	0	0	0
$S_5$	0.17	0	0	0	0	0.26	0	0.21	0	0	0	0	0.36	0	0	0
$S_6$	0	0.36	0	0	0.26	0	0.17	0	0	0	0	0	0	0	0	0.21
$S_7$	0	0	0.26	0	0	0.17	0	0.36	0	0	0	0	0	0	0.21	0
$S_8$	0	0	0	0.26	0.21	0	0.36	0	0	0	0	0	0	0.17	0	0
$S_9$	0.21	0	0	0	0	0	0	0	0	0.26	0	0.36	0.17	0	0	0
$S_{10}$	0	0	0	0.17	0	0	0	0	0.26	0	0.21	0	0	0.36	0	0
$S_{11}$	0	0	0.36	0	0	0	0	0	0	0.21	0	0.26	0	0	0.17	0
$S_{12}$	0	0.21	0	0	0	0	0	0	0.36	0	0.26	0	0	0	0	0.17
$S_{13}$	0	0	0	0	0.36	0	0	0	0.17	0	0	0	0	0.21	0	0.26
$S_{14}$	0	0	0	0	0	0	0	0.17	0	0.36	0	0	0.21	0	0.26	0
$S_{15}$	0	0	0	0	0	0	0.21	0	0	0	0.17	0	0	0.26	0	0.36
$S_{16}$	0	0	0	0	0	0.21	0	0	0	0	0	0.17	0.26	0	0.36	0

Table 3.1 : Probabilités des messages confidentiels dans un réseau de communication de 16 experts.

Pour appliquer notre implimentation au problème de la table 3.1, nous allons multiplier les probabilités par cent afin d'avoir des valeurs entières. Notre application programmée sur le code blocks nous a donné l'arbre couvrant représenté par les liens de la figure 3.13.



```
C:\Users\dell\Downloads\pri.exe
Edge  Weight
0 - 1  26
1 - 2  17
2 - 3  21
0 - 4  17
6 - 5  17
14 - 6  21
4 - 7  21
0 - 8  21
3 - 9  17
9 - 10  21
1 - 11  21
8 - 12  17
7 - 13  17
10 - 14  17
11 - 15  17

Process returned 0 (0x0)   execution time : 0.500 s
Press any key to continue.
```

FIGURE 3.13 – Résultat.

L'arbre couvrant associé aux liens obtenus par notre application est donné par la figure 3.14.

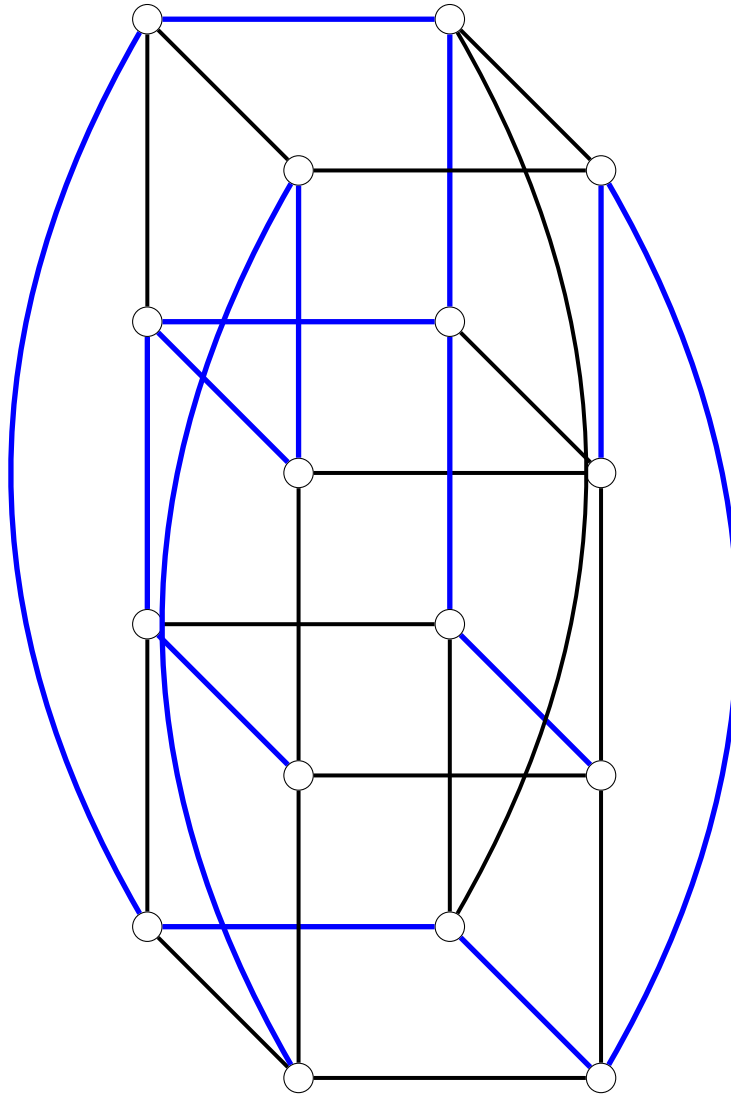


FIGURE 3.14 – Arbre couvrant de poids minimum.

*La probabilité minimal de l'arbre est  $P^* = 1 - \prod_{i,j \in T} (1 - P_{ij}) = 0.959$ .*

## Conclusion

L'importance de la théorie des graphes est de permettre de modéliser un nombre impressionnant de problèmes auxquels nous sommes quotidiennement confrontés, on peut citer : problèmes de transport, de routage dans les réseaux, d'emploi du temps, de gestion de projets, allocation des fréquences, etc. Le problème de coloration des graphes est un domaine très actif de la théorie des graphes du fait de ses nombreuses applications.

Dans ce manuscrit, nous avons après avoir rappelé les notions de base de théorie des graphes et la coloration propre des arêtes des graphes. Nous avons caractérisé des nouvelles classes d'arbres binaires pour lesquelles nous avons donné l'indice chromatique qui vérifie la propriété de parité, c'est-à-dire, la coloration des arêtes de l'arbre par les entiers de l'ensemble  $\{1, \dots, k\}$  tel que pour toute chaîne  $P$  de l'arbre il existe  $i \in \{1, \dots, k\}$  pour le quelle un nombre impair de  $P$  sont marquées par  $i$ , et nous avons vérifiée une deuxième propriété qui dit que pour tout cycle  $C$  appartient à l'arbre  $\forall i \in \{1, \dots, k\}$ , le nombre d'arêtes de  $C$  coloriées par  $i$  est pair, car les deux classes que nous avons caractérisé sont bipartis.

Par ailleurs, nous avons montré que chacune de ces deux nouvelles classes est plongeable dans son hypercube optimal.

A la fin, nous avons implémenté d'algorithme de Prim qui consiste à déterminer l'arbre couvrant de poids minimum afin de montrer l'importance de plongement d'arbre dans l'hypercube  $Q_n$ . Nous avons présenté un problème concret pour le quel la solution donnée sous forme d'un arbre plongeable dans l'hypercube  $Q_n$ .

Comme perspective, nous allons essayer de :

- Traiter des nouvelles classes de graphe  $G$  non acyclique pour lesquelles nous allons calculer  $\hat{P}(G)$  et de déterminer le plus petit hypercube qui contient chacune d'elles.
- Programmer l'algorithme de Kruskal et l'appliquer au problème précédent puis comparer les résultats obtenus.



## Bibliographie

- [1] A. Bondy and U.S.R. Murty. <<Graph theory>>. (2008).
- [2] A. Bretto et A. Faisant et F. Hennecart. Éléments de théorie des graphes, (2012).
- [3] A. Chetbani et H. Berdous et B. Benabbou et S.Boudjelda. La recherche des points d'articulations dans un réseau de télécommunication :Cas réseau Fibre-Optique Algérie Télécom Béjaia, mini projet, Université Adberrahmane Mira Bejaia, (2016).
- [4] B. Brahmi, Cours Master 2, Recherche Opérationnelle , (2017-2018).
- [5] B. Garcia, cour Recherche Opérationnelle, (2000).
- [6] B. Benmedjdoub. Colorations d'incidences, (2018).
- [7] C. Berge. Graphes et hypergraphes. Dunod, edition Paris, (1970).
- [8] C. Berge. Two theorems in graph theory. Proceedings of the National Academy of Sciences. 43, pp. 842-844, (1957).
- [9] D. Yadolah . Optimisation appliquée ,Editeur : Springer Livre ,2005.
- [10] D. Muller .Introduction à la théorie des graphes. (2008).
- [11] D. P. Bunde et all. Parity and Strong Parity Edge-Coloring of Graphs. Congr Numer. 187, pp. 193-213, (2007).
- [12] I. Havel and P. Liebl. Embedding the dichotomic tree into the n-cube. Časpis Pěst. Mat. 97, No. 2, pp. 201-205, (1972).
- [13] I. Havel. On certain trees in hypercubes. Topics in combinatorics and graph theory. pp. 353-358, (1990).
- [14] J. cohen. Théorie des graphes et algorithmes. (2006).
- [15] K. Kabyl, A. Berrachedi and E. Sopena : A note on the cubical dimension of new classes of binary trees. Czechoslovak Mathematical Journal, 65 (140), pp. 151-160, (2015).
- [16] K. Benjamin. Modélisation, Optimisation, Complexité et algorithme. (2005-2006).
- [17] L. Jacques. Théorie des graphe, Modulo Editeur, Québec, (1981).
- [18] M. Kobeissi. Plongement de graphe dans l'hypercube, thèse de doctorat, disipline informatique. Université Josef Fourier, (2001) Grenoble.

- [19] M. Gandron and M. Minoux. Graphes et algorithmes. Editions Eyrolles, Paris VI, (1986).
- [20] N. SBIHI. Contribution a l'étude des stables dans un graphe par une approche algorithmique, thèse doctorat, université Josef Fourier (Grenoble), (1987).
- [21] S. Taouinet, cours théorie des graphes avancée Master1.
- [22] S. Foldes. A characterisation of hypercubes. Discrete Math. 17, pp. 155-159, (1977).
- [23] S. Skiena. The Algorithm Design Manual, Springer, 2ème édition, (2008).
- [24] S. YAHIAOUI et N. MAOUNI. Quelques techniques d'optimisation dans les réseaux et applications, mémoire master2. Université de Abderrahmane Mira Bejaia, (2018).
- [25] V.G. Vizing. On an estimate of the chromatic class of a p-graph. Metody Diskret.Anal. 3 :25-30, (1964).
- [26] V.I. Volishin. introduction to graph theory, Nova Science Publishers, New york, (2009).

## **Résumé**

*Le but principal de ce manuscrit est d'étudier la parité et forte parité de l'indice chromatique de nouvelles classes d'arbres binaires. Un problème d'une nature sera traité, il concerne le plongement de ces arbres dans l'hypercube optimal (hypercube de dimension minimale). Dans notre travail, nous avons introduit inductivement deux nouvelles classes d'arbres binaires pour lesquelles  $P$  et  $\hat{P}$  sont déterminés. A la fin, nous avons implémenté l'algorithme de Prim sur code blocks*

**Mots clés :** *Graphe, arbres, hypercube, coloration, indice chromatique.*

## **Abstract**

*The main of this manuscript is to study the parity and strong parity of edge coloring of new classes of binary trees. Another problem will be treated, it concerns the embedding of these trees in the optimal hypercube (hypercube of dimension minimal). In our work, we have inductively introduced two new classes of binary trees for which we determined  $P$  and  $\hat{P}$ . Finally we have implemented Prim's algorithm on the code blocks.*

**Keywords :** *graph, trees, hypercube, coloring, chromatic index.*