

Graphes.

Jean-Paul Davalan

1 Exemples introductifs, concept de graphe.

1.1 Exemples de 1-graphes orientés.

Les graphes au programme¹ sont les 1-graphes orientés finis dont les premiers exemples vont de la figure [fig. 1] à la figure [fig. 8].

Lorsqu'il y a au plus un *arc* d'un *sommet* vers un autre, on dira que le graphe est un 1-graphe orienté.

1.1.1 Division dans $[1; 10] \cap \mathbb{N}$.

Soit le sous-ensemble $S = \{1; 2; 3; 4; \dots; 10\}$ de l'ensemble \mathbb{N}^* des nombres entiers naturels non nuls et $A \subset S \times S$ l'ensemble des couples (x, y) tels que $x \in S$ soit un diviseur de $y \in S$.

Par exemple $(3; 6) \in A$ et $(3; 7) \notin A$.

On peut représenter la situation par la figure [fig. 1].

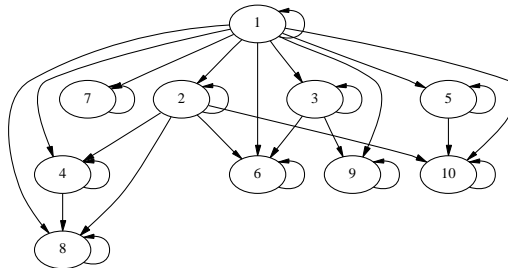


FIG. 1 – Graphe de la relation « est diviseur de ».

Appelons G le graphe, S l'ensemble des *sommets* et A l'ensemble des *arcs* du graphe G .

Un arc a deux *extrémités*, une *extrémité initiale* et une *extrémité terminale*.

Un arc dont les deux extrémités sont confondues est une *boucle*.

Un *pseudo-graphe* est un graphe qui a des boucles.

Un 1-graphe qui n'a pas de boucle s'appelle un *graphe simple*.

Dans le cas d'un 1-graphe, l'ensemble A des arcs est assimilé à un sous-ensemble de $S \times S = S^2$.

Le graphe s'écrit $G = (S, A)$.

¹Le programme des classes de BTS se trouve à l'annexe A.

Le nombre de sommets du graphe est appelé l'*ordre* du graphe. (Par exemple l'ordre du graphe des diviseurs, donné en exemple, est 10).

Le *degré* d d'un sommet x est le nombre d'arcs dont l'une des extrémités est x (on note d^+ le nombre d'arcs qui partent de x et d^- le nombre d'arcs qui arrivent en x , $d = d^- + d^+$, d^- est le degré entrant, d^+ est le degré sortant).

Exercices :

1. Montrer que la somme des degrés de tous les sommets est le double du nombre d'arcs du graphe.
2. Montrer que le nombre des sommets de degrés impairs est pair.
3. Montrer que le nombre d'arcs d'un 1-graphe (sans boucles) d'ordre n est inférieur ou égal à $n(n-1)$.
4. Quel est le nombre d'arcs lorsque les n sommets du graphe ont même degré $d = k$?

1.1.2 Les trois villas.

Le problème est le suivant : D'un même côté d'une route il y a trois maisons et de l'autre côté trois usines, d'eau, de gaz et d'électricité, (figure [fig. 2]). Pouvez-vous tracer de chacune des trois usines, trois conduites d'alimentation aux trois villas de telle manière qu'aucune des neuf conduites n'en coupe une autre ?

Ce problème n'a pas de solution, l'explication utilise la relation d'Euler $s + f - a = 2$ reliant les nombres s de sommets, f de faces et a d'arêtes, relation vérifiée par tout graphe planaire connexe. (Les arêtes ou arcs ne se coupent pas, une face est une région de la figure, la face externe doit être comptée).

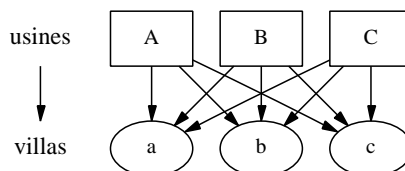


FIG. 2 – Les trois villas.

L'ensemble des sommets du graphe G est $S = \{A, B, C, a, b, c\}$, l'ensemble des arcs est $\Gamma = \{A, B, C\} \times \{a, b, c\}$, le graphe est donc $G = (S, \Gamma)$, (figure [fig. 3]).

L'ensemble des *successeurs* du sommet A est $\{a, b, c\}$, l'ensemble de ses *prédécesseurs* est vide. L'ensemble des prédécesseurs de a est $\{A, B, C\}$.

On remarque que l'ensemble $X = \{A, B, C, a, b, c\}$ des sommets est la réunion de deux parties $\{A, B, C\}$, $\{a, b, c\}$ disjointes et non vides, telles que deux sommets d'une même partie ne sont jamais reliés par un arc. On dit que ce graphe est un *graphe biparti*.

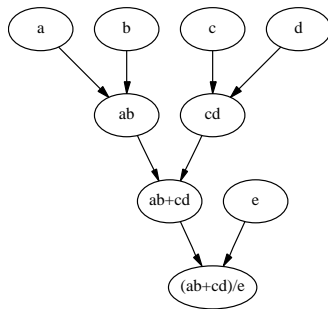


FIG. 3 – Calcul de $(ab + cd)/e$.

1.1.3 Un arbre algébrique.

Exercice :

- Dessiner un graphe correspondant à $3x^2 e^{x/\ln x}$. Est-ce un arbre ?

1.1.4 Un algorithme.

Soit $X = \{1; 2; \dots ; N\}$ où N est un entier donné. L'algorithme est le suivant : si $x \in X$ est pair, alors x a pour successeur $x/2$, sinon si $x + 2$ est multiple de 3, alors x a pour successeur $(x + 2)/3$, enfin si aucune des conditions précédentes n'est remplie, x a pour successeur $x - 1$. Voir la figure [fig. 4].

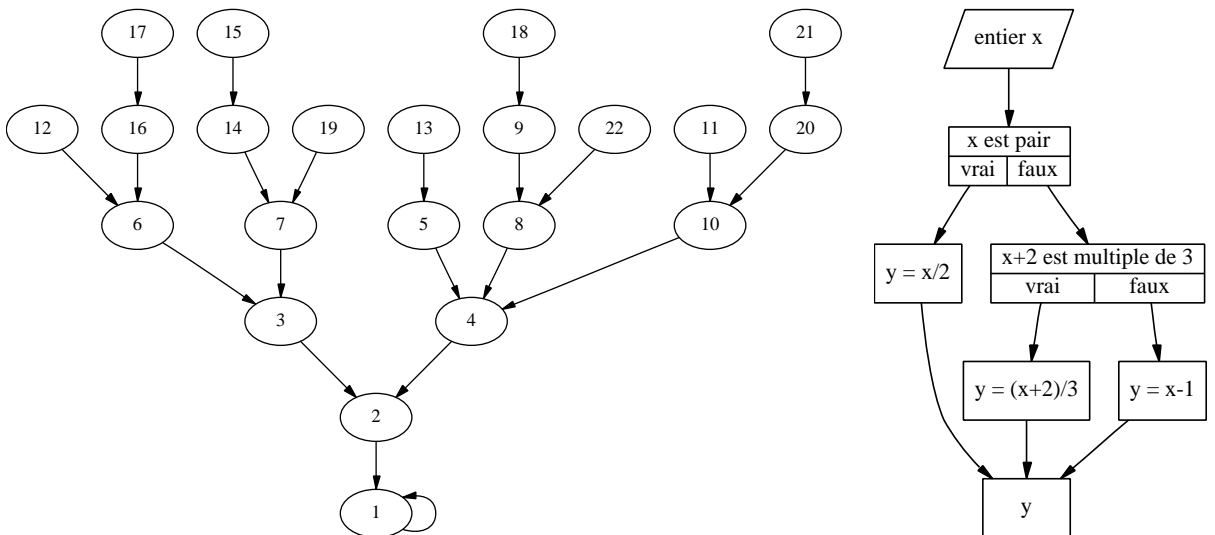


FIG. 4 – Graphes et algorithmes.

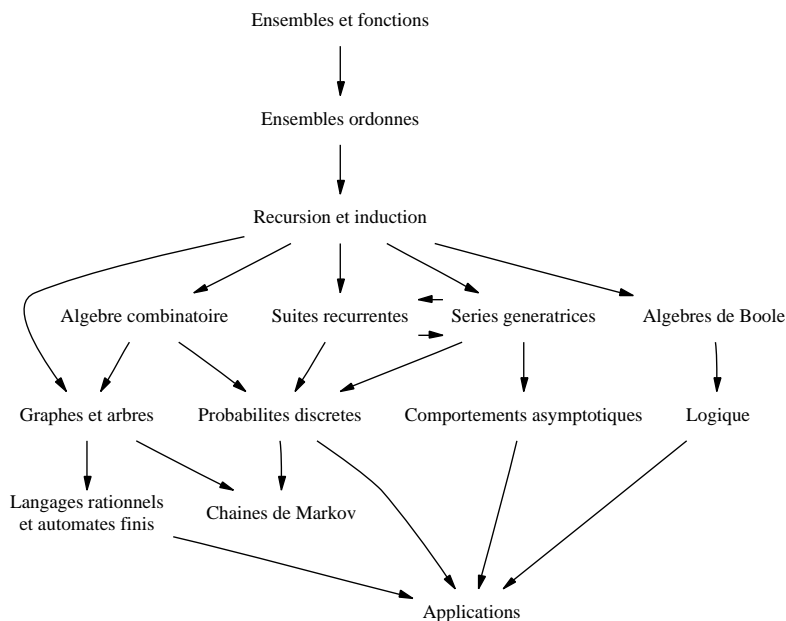


FIG. 5 – *Interdépendances des chapitres.* [13]

La conjecture de Collatz² : Dans le même genre d'idées, on prend sur l'ensemble infini \mathbb{N}^* de sommets, l'algorithme suivant,
 « si x est pair, il a pour successeur $x/2$, sinon s'il est impair, son successeur est $(3x + 1)/2$ ». Voir [16] pour de plus amples détails sur la conjecture de Collatz. Dessiner une partie du graphe. Peut-on trouver un ou plusieurs circuits dans le graphe ? Quelle conjecture pouvez-vous formuler ?

1.1.5 Interdépendance.

Le graphe de la figure [fig. 5] est extrait de l'introduction du livre [13] « Mathématiques pour l'informatique ».

1.2 Exemples de multigraphes.

1.2.1 Un 3-graphe.

Le graphe représenté sur la figure [fig. 6] est un 3-graphe car le nombre maximum d'arcs reliant un sommet à un second sommet est inférieur ou égal à 3 (les arcs numérotés 1, 2 et 3 correspondent au même couple (a, b) de deux sommets). À un couple peut correspondre plusieurs arcs, une définition plus générale que la précédente est nécessaire.

²La conjecture de Collatz est aussi connue comme problème de Syracuse, de Kakutani, de Hasse, de Ulam ou problème $3x + 1$, $(3x + 1)$ problem en anglais).

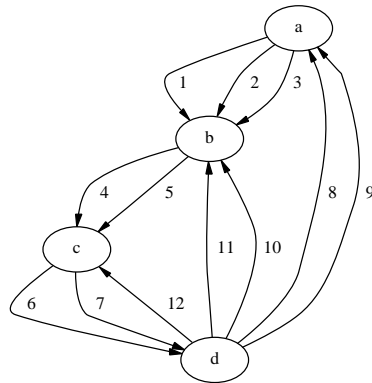


FIG. 6 – Arcs multiples.

1.3 Définitions.

Définition 1.1 Un graphe orienté est un quadruplet $\Gamma = (X, A, \alpha, \beta)$ où

- 1 — X est un ensemble de sommets,
- 2 — A est un ensemble d'arcs, disjoint de X , ($X \cap A = \emptyset$),
- 3 — α et β sont deux applications de A dans X telles que pour tout arc a de A
 - $\alpha(a)$ est l'origine (ou prédécesseur) de x ,
 - $\beta(a)$ est le but (ou successeur) de x .

Définition 1.2 Un p -graphe est un graphe où p arcs au plus ont même prédécesseur et même successeur.

1.3.1 Chemin, chaîne, cycle, circuit.

Dans un 1-graphe, un *chemin* est une liste ordonnée (x_1, x_2, \dots, x_p) de sommets telle que pour tout i , $1 < i < p$, il existe un arc allant de x_i à x_{i+1} . Une chaîne qui ne rencontre pas deux fois le même sommet est un *chemin élémentaire*, une chaîne qui n'utilise pas deux fois le même arc est un *chemin simple*. Une définition d'un *chemin* adaptée aux multigraphes utilise une liste ordonnée d'arcs.

Définition 1.3 Un *chemin* est une liste ordonnée d'arcs (a_1, a_2, \dots, a_p) telle que l'extrémité de a_i est l'origine de a_{i+1} .

Si l'on ne tient pas compte de l'orientation de l'arc, on a une *chaîne*.

Définition 1.4 Une *chaîne* est liste ordonnée d'arcs (a_1, a_2, \dots, a_p) telle que pour tout $1 < i < p$, l'arc a_i a une extrémité commune avec a_{i-1} et l'autre extrémité commune avec a_{i+1} .

Définition 1.5 Un *cycle* est une chaîne qui ne contient pas deux fois le même arc et telle que les sommets aux deux extrémités de la chaîne coïncident.

Un *circuit* est un cycle (a_1, a_2, \dots, a_p) tel que l'extrémité terminale de a_i soit égale à l'extrémité initiale de a_{i+1} pour tout i , (si $i = p$, on prendra 1 au lieu de $p + 1$: l'extrémité terminale de a_p doit être égale à l'extrémité initiale de a_1).

Comme les cycles, les circuits ne contiennent pas deux fois le même arc. (Sinon il s'agit d'un pseudo-cycle ou d'un *pseudo-circuit*). Voir la figure [fig. 7].

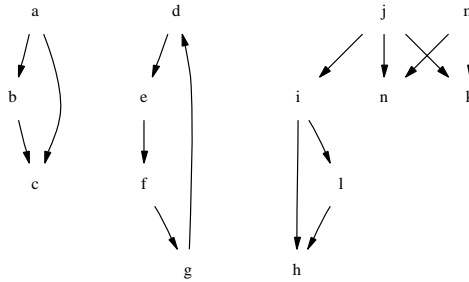


FIG. 7 – Cycle, circuit, pseudo-cycle.

1.3.2 Graphe connexe, graphe non-connexe.

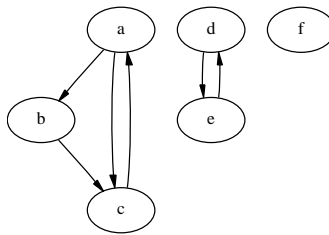


FIG. 8 – Graphe non-connexe.

Définition 1.6 Un graphe connexe est un graphe tel que pour toute paire de deux sommets distincts x et y , il existe une chaîne dont l'une des extrémités est x , l'autre extrémité étant y .

S'il existe deux sommets qui ne peuvent être reliés par aucune chaîne, le graphe n'est pas connexe, il est en plusieurs parties qui, elles, sont connexes : ce sont les *composantes connexes* du graphe.

le graphe de la figure [fig. 8] a trois composantes connexes.

1.3.3 Graphe non orienté.

Dans certains cas, l'orientation des arcs n'est pas utile, on ne tient plus alors compte de la distinction origine - but des deux extrémités.

Définition 1.7 Un graphe non orienté est un triplet $\Gamma = (X, A, \gamma)$ où γ est une application qui associe à chaque arête $a \in A$ l'ensemble de ses deux extrémités $\gamma(a) = \{x_1, x_2\}$. (Si $x_1 = x_2$, l'arête a est une boucle).

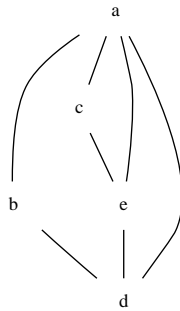


FIG. 9 – Graphe non-orienté.

2 Représentations des graphes.

2.1 Schémas et tables

2.1.1 Diagramme sagittal.

Les figures [fig. 1] à [fig. 8] où les arcs sont représentés à l'aide de flèches sont des schémas sagittaux. Par extension, on pourra appeler diagramme sagittal la figure [fig. 9].

2.1.2 Tableau.

Différents types de tableaux peuvent être utilisés (voir la figure [fig. 10]) suivant l'usage désiré, présentation, classification ou autre.

2.1.3 Diagramme cartésien.

Dans un diagramme cartésien d'un 1-graphe, (voir la figure [fig. 10]), les sommets sont représentés sur les deux axes des abscisses (prédécesseurs) et des ordonnées (successeurs). Les points de coordonnées $(x; y)$ du diagramme cartésien correspondent aux arcs de prédécesseur x et de successeur y .

2.2 Matrices associées ou matrices d'adjacence.

2.2.1 Définition et exemple de matrice d'adjacence.

Soit un 1-graphe $G = (X, U)$ d'ordre n et de p arcs, l'ensemble des n sommets est X , la famille des p arcs est U .

Notons x_1, x_2, \dots, x_n les n sommets, donc sous la forme d'une famille finie indexée par les éléments de $\{1; 2; \dots; n\}$, ce qui revient à ordonner les n sommets.

La matrice d'adjacence A du graphe est alors une matrice carrée d'ordre n , de n lignes et n colonnes, obtenue en écrivant un 1 à l'intersection de la ligne i et de la colonne j lorsqu'il existe un arc de prédécesseur x_i et de successeur x_j et en écrivant un 0 s'il n'existe pas d'arc.

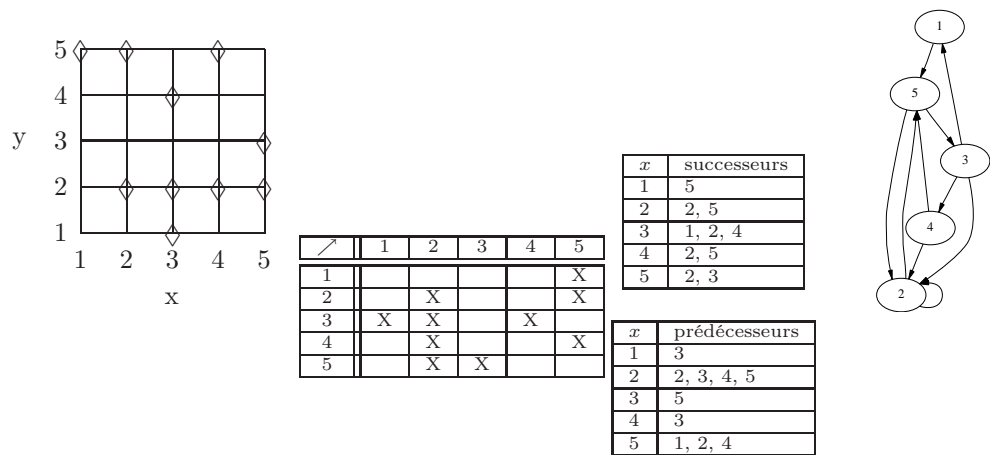


FIG. 10 – Diagramme cartésien, tableaux, diagramme sagittal.

$$a_{i,j} = \begin{cases} 0 & \text{s'il n'existe pas d'arc de } x_i \text{ vers } x_j, \\ 1 & \text{s'il existe un arc de } x_i \text{ vers } x_j, \end{cases} \quad (1)$$

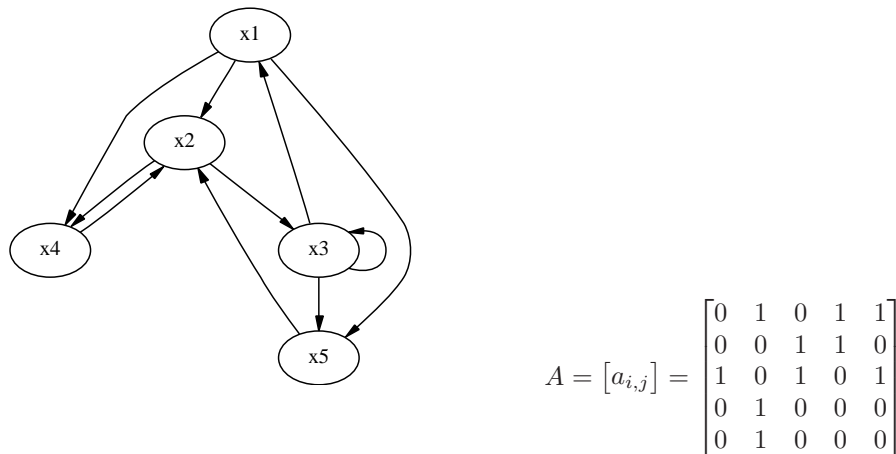


FIG. 11 – Matrice d'adjacence.

2.2.2 Matrices booléennes.

Définition 2.1 Une matrice $A = [a_{i,j}]$ est booléenne si tous ses éléments $a_{i,j}$ sont égaux à 0 ou à 1.

Les matrices associées aux graphes d'ordre n (de n sommets) sont donc des matrices booléennes d'ordre n (de n lignes et n colonnes).

Addition et produit de matrices booléennes. Soient $A = [a_{i,j}]$ et $B = [b_{i,j}]$ deux matrices carrées booléennes d'ordre n .

Somme.

La somme booléenne $A \oplus B$ des deux matrices a et B est la matrice $C = [c_{i,j}]$ dont chacun des éléments $c_{i,j}$ est obtenu en effectuant la somme booléenne de $a_{i,j}$ et de $b_{i,j}$, $c_{i,j} = a_{i,j} \oplus b_{i,j}$.

$$\text{Par exemple } \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix} \oplus \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 0 \end{bmatrix}.$$

$$\text{Les propriétés des deux matrices particulières } \Omega = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \text{ et } U = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

sont évidentes : $A + \Omega = A$ et $A + U = U$.

Ω est la matrice nulle, élément neutre de l'addition.

U est absorbante.

Produit.

Le produit booléen s'effectue comme le produit habituel sauf que les sommes et produits de réels sont remplacés par des sommes et produits booléens.

$$\text{Par exemple } \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix} \otimes \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 0 & 0 \end{bmatrix}.$$

La matrice $I = [a_{i,j}]$

$$a_{i,j} = \begin{cases} 0 & \text{si } i \neq j, \\ 1 & \text{si } i=j, \end{cases} \quad (2)$$

est élément neutre du produit booléen.

Puissance.

Une puissance de matrice booléenne sera notée $A^{[n]} = \underbrace{A \otimes A \otimes \dots \otimes A}_{n \text{ facteurs}}$.

$$\text{Par exemple, si } A = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \text{ alors } A^{[2]} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix} \otimes \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix} =$$

$$\begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 0 \end{bmatrix} \text{ et } A^{[3]} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix} \otimes \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 0 \end{bmatrix}. \text{ On en déduit que}$$

$$\text{pour tout } n \geq 2, A^{[n]} = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 0 \end{bmatrix}.$$

3 Problèmes de recherche de chemins.

3.1 Problème du chemin entre deux points.

3.1.1 Exemples classiques.

Le « problème du chemin » est de trouver (le plus rapidement possible) un chemin d'un sommet donné a à un sommet donné b .

Exemple : Le problème du loup, de la chèvre et du chou.

Une chèvre, un loup et un chou se trouvent sur la rive droite d'un fleuve. Un passeur veut leur faire traverser le fleuve, mais sa barque étant trop petite, il ne peut transporter qu'un seul d'entre eux à la fois. On ne peut laisser sans surveillance le loup en compagnie de la chèvre ou la chèvre en compagnie du chou. Expliquez au passeur ce qu'il peut faire.

Le graphe [fig. 12] ne comporte que les états admissibles, sur la rive droite. Par exemple CP correspond à la situation « chèvre et passeur ensemble sur la rive droite » qui est autorisée car le loup et le chou se trouvent ensemble sur l'autre rive, ce qui est sans risque.

Pour résoudre le problème il suffit de trouver un chemin de l'état initial CLXP à l'état final « vide ».

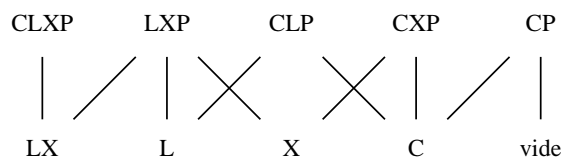


FIG. 12 – Chèvre, loup, chou et passeur : C L X P.

Exemple : Le labyrinthe.

Un labyrinthe peut être représenté par un graphe *planaire* ([fig. 13] par exemple). On suppose que les galeries sont sur un même niveau, un même plan, elles ne peuvent se croiser sans se rencontrer (elles ne peuvent pas se superposer). Comment sortir ? Trouver une méthode générale, utilisable dans tout labyrinthe, qui permet de trouver la sortie.

3.2 Hamilton ou Euler ?

3.2.1 Chemins hamiltoniens.

Définition 3.1 On dit qu'un chemin est hamiltonien s'il passe une fois et une seule par chaque sommet du graphe.

Exemple 1 : Voyage autour du monde (Hamilton³).

Donnons-nous 20 villes sur le globe terrestre. On se propose de passer une fois et une seule par chacune de ces villes et de revenir au point de départ (voir la figure [fig. 14]).

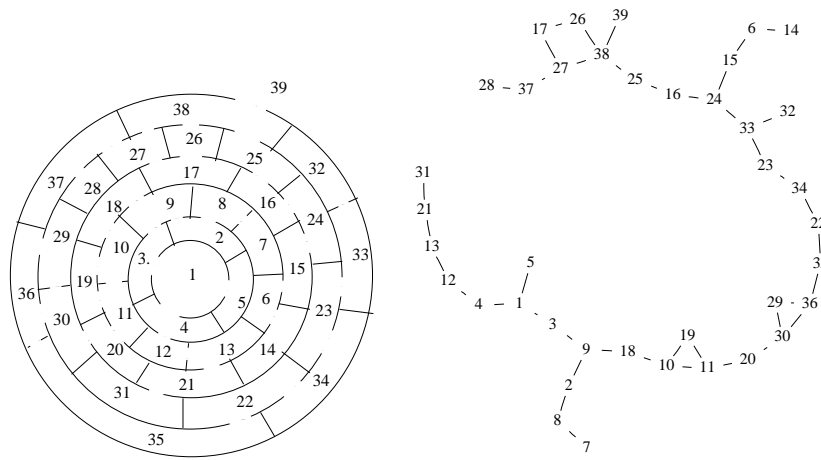


FIG. 13 – *Labyrinthe et son graphe.*

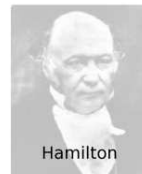
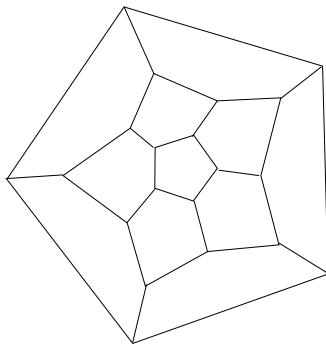


FIG. 14 – *Voyage autour du monde (Hamilton).*

Exemple 2 : Le parcours du cavalier sur l'échiquier (Euler⁴).

Le cavalier doit parcourir l'échiquier en passant une fois et une seule sur chacune des cases. Prendre un échiquier ou utiliser un programme (voir quelques unes des solutions à la figure [fig. 15] ou le programme récursif de la figure [fig. 16] qui est adaptable à d'autres recherches de chemins hamiltoniens).

Exemple 3 : (Casse-tête). Presque identique à la course du cavalier, sauf que le déplacement est de 3 cases horizontalement ou verticalement, ou encore de 2 cases en diagonale. Voir trois solutions à la figure [fig. 17].

Il suffit d'écrire la ligne `#define DIF dif_ct // ct: casse-tête` à la place de la quatrième ligne
`#define DIF dif_cv // cv: cavalier` du programme de la figure [fig. 16]

³Sir William Rowan Hamilton, né le 4 Août 1805 et décédé le 2 Septembre 1865 à Dublin, Irlande.

⁴Leonhard Euler, né le 15 Avril 1707 à Bâle (Suisse), décédé le 18 Septembre 1783 in St. Petersbourg, Russie.

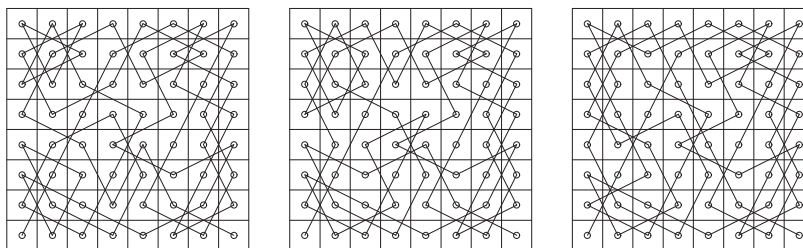


FIG. 15 – Course du cavalier (quelques solutions).

```

#include<stdio.h>
#include <stdlib.h>
#include <time.h>
#define DIF dif_cv // cv: cavalier
#define NMAX 12 // nbre MAX de cases horiz. et vert.

int N = 5, NN, // nbre MAX de cases horiz. et vert.
    num = 0, // nbre de solutions
    quiet = 0, // 1 n'affiche pas
    quad[NMAX][NMAX]; // échiquier
/* déplacements relatifs en x ou en y */
int dif_ct[8][2] = /* casse-tête */
{{3,0},{2,2},{0,3},{-2,2},{-3,0},{-2,-2},{0,-3},{2,-2}};
int dif_cv[8][2] = /* déplacements du cavalier */
{{2,1},{2,2},{-1,2},{-2,1},{-2,-1},{-1,-2},{1,-2},{2,-1}};

void solution()
{
    int i, j;
    printf("# %d\n", num);
    for (i = 0; i < N; i++) {
        printf(" ");
        for (j = 0; j < N; j++) printf("%3d", quad[i][j] - 1);
        printf("\n");
    }
    printf("\n");
}

void next(int n, int x, int y) /* position suivante */
{
    int k, X, Y;

    quad[x][y] = n;
    if (n < NN) {
        for (k = 0; k < 8; k++) {
            X = x + DIF[k][0], Y = y + DIF[k][1];
            if (X>0 && X<N && Y>=0 && Y<N && quad[X][Y]==0)
                next(n + 1, X, Y);
        }
    } else if (n == NN) /* affichage de la solution */
        num++;
    if (!quiet || num <= 1) solution();
}

int main(int argc, char *argv[])
{
    int i, j;
    long long c_begin = (long long)clock();
    if (argc > 1) N = atoi(argv[1]);
    N = (N <= NMAX) ? N : 5;
    NN = N * N;
    if (argc > 2 && argv[2] == 'q') quiet = 1;
    printf("# Solutions du casse-tête %dx%d\n", N, N);
    for (i = 0; i < N; i++)
        for (j = 0; j < N; j++) quad[i][j] = 0;
    next(1, 0, 0);
    printf("# %d solutions (%Ld s)\n\n",
        num, (clock() - c_begin) / CLOCKS_PER_SEC);
    return 0;
}

```

FIG. 16 – Course du cavalier (programme).

pour que le programme modifié permette d'obtenir les très nombreuses solutions du casse-tête (prévoir une durée de quelques siècles ou millénaires dans le cas d'un échiquier de 8×8 cases).

Exemple 4 : Sur un échiquier de 8×8 cases, chercher un chemin de la case (1, 1), en bas à gauche, à la case (8, 8), en haut à droite, en passant une fois et une seule par toutes les cases de l'échiquier et en se déplaçant comme une « tour » qui avance d'une seule case à la fois, c.-à-d. d'une case horizontalement ou verticalement mais pas en diagonale.

Exemple 5 : Le graphe de Petersen⁵ (voir la figure [fig. 18]) ne contient pas de cycle hamiltonien mais l'élimination d'un sommet quelconque donne un graphe avec cycle hamiltonien.

3.2.2 Chaînes eulériennes.

Euler est avec Leibniz un précurseur de la théorie des graphes et de la topologie.

⁵Mathématicien danois. Voir [29].

| # 3855 | # 4063 | # 5008 |
|-------------------------|-------------------------|-------------------------|
| 0 40 43 7 18 44 6 23 | 0 40 43 7 18 44 6 23 | 0 39 55 7 18 54 6 23 |
| 33 63 48 34 13 47 35 12 | 29 63 56 34 13 55 35 12 | 29 63 35 30 13 36 31 12 |
| 42 51 25 41 50 24 17 45 | 42 47 25 41 46 24 17 45 | 56 43 25 38 33 24 17 53 |
| 1 39 32 8 19 31 5 22 | 1 39 30 8 19 33 5 22 | 1 40 48 8 19 51 5 22 |
| 28 62 49 37 14 46 36 11 | 28 62 57 37 14 54 36 11 | 28 62 34 42 14 37 32 11 |
| 59 52 26 60 55 9 16 56 | 59 48 26 60 51 9 16 52 | 57 44 26 60 47 9 16 52 |
| 2 38 29 3 20 30 4 21 | 2 38 31 3 20 32 4 21 | 2 41 49 3 20 50 4 21 |
| 27 61 58 53 15 57 54 10 | 27 61 58 49 15 53 50 10 | 27 61 58 45 15 59 46 10 |

FIG. 17 – Trois solutions du casse-tête.

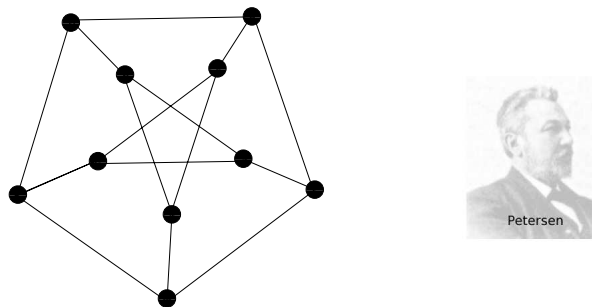


FIG. 18 – Graphe de Petersen. Julius Peter Christian Petersen, 1839-1910, Danemark.



FIG. 19 – Gottfried Wilhelm von Leibniz, 1646-1716, Allemagne.

Définition 3.2 On appelle chaîne eulérienne (resp. cycle eulérien) une chaîne (resp. un cycle) qui emprunte une fois et une seule chaque arête du graphe.

Exemple 1 : (voir [fig. 20]).

Est-il possible de tracer l'un de ces dessins sans soulever le crayon ni passer deux fois sur le même trait. (Et sans replier la feuille).

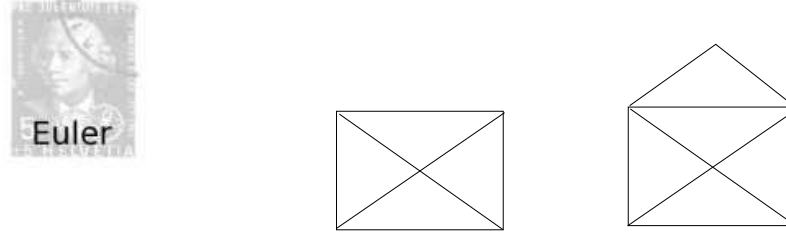


FIG. 20 – *Cycles eulériens ? (Euler).*



FIG. 21 – *Kœnigsberg.*

Exemple 2 : Les ponts de Kœnigsberg.

En 1736 les habitants de Kœnigsberg (aujourd'hui Kaliningrad), se passionnaient, sans parvenir à le résoudre, pour un problème (Voir [fig. 21] et [fig. 22]) :

Un piéton, en se promenant, peut-il traverser une fois et une seule chacun des sept ponts qui enjambent la rivière Pregel ?

Exemple 3 : Codes d'accès.

Vous avez oublié le code de $p = 3$ lettres prises parmi les $n = 4$ lettres a, b, c ou d qui permet d'ouvrir la porte de votre immeuble. Quel mot, le plus court possible, contient tous les codes ? (Dans la réalité le code est sûrement de plus de trois caractères à choisir dans un alphabet de plus de quatre lettres, mais pour pouvoir dessiner un graphe lisible [fig. 23], il est nécessaire de simplifier).

Les mots cherchés s'appellent des suites de *Debruijn*. Il existe plusieurs solutions, toutes de même longueur minimale p^n . Pour en obtenir un, il suffit de trouver un circuit eulérien dans le graphe (figure [fig. 23]) dont les sommets

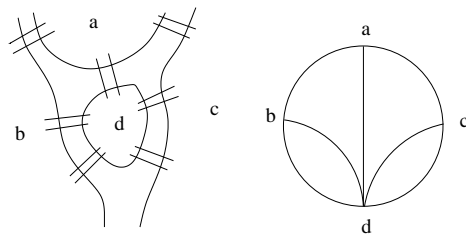


FIG. 22 – La Pregel et l'île de Kneiphof à Kœnigsberg.

sont aa, ab, \dots et dont les arcs sont les mots de trois lettres. Ainsi l'arc nommé abd a pour origine ab et pour but bd .

Une solution est le mot de $4^3 = 64$ lettres

abaacaadabbabcabdacbccadadbadcaddbbcbdbccbdcbddcccdccddaa,

le chemin eulérien correspondant emprunte les arcs aba, baa, aac, \dots et les sommets successifs sont ab, ba, aa, ac, \dots pour arriver en aa et enfin revenir au sommet initial ab .

Comme tous les sommets s du graphe ont un degré $d(s)$ pair et sont tels que $d^+(s) = d^-(s)$, le problème a une solution. (Chacun des sommets est l'extrémité et l'origine d'un même nombre d'arcs ou encore : il y a exactement autant d'arcs qui arrivent en chacun des sommets que d'arcs qui repartent des mêmes sommets).

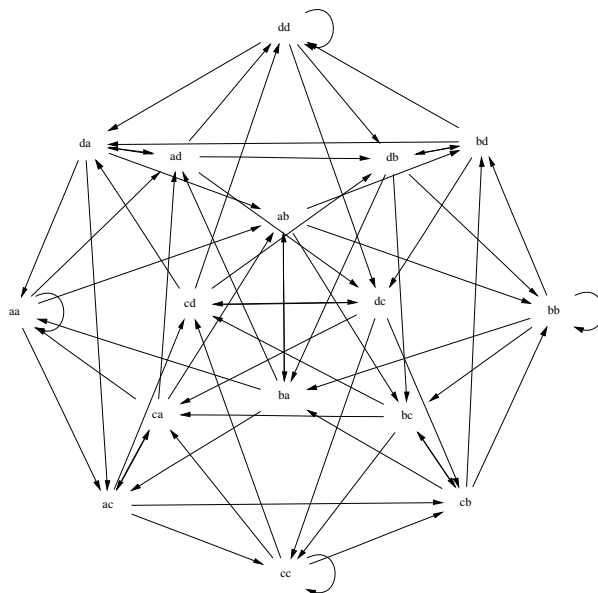


FIG. 23 – Codes d'accès.

3.3 Chemins de longueur donnée.

3.3.1 Longueur d'un chemin.

Définition 3.3 La longueur d'un chemin $c = s_1 s_2 \dots s_p$ de p sommets est le nombre des arcs utilisés, c'est-à-dire $p - 1 : l(c) = p - 1$.

3.3.2 Existence d'un chemin de longueur donnée.

Exemple : Sur la figure [fig. 24], pouvez-vous trouver des chemins de longueur 1, 2, ... du sommet a au sommet b ?

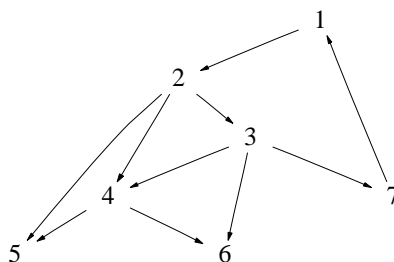


FIG. 24 – Longueurs de chemins.

$$\begin{array}{ccc}
 M^{[1]} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} & M^{[2]} = \begin{bmatrix} 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} & M^{[3]} = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 \end{bmatrix} \\
 M^{[4]} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix} & M^{[5]} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix} & M^{[6]} = \begin{bmatrix} 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}
 \end{array}$$

La matrice d'incidence $M = M^{[1]}$ et les puissances booléennes de M ci-dessus permettent de déterminer l'existence d'un chemin de longueur p d'un sommet x_i à un sommet x_j :

L'élément $a_{i,j}$ de la matrice $M^{[p]}$ vaut

$$\begin{cases} 0 & \text{s'il n'existe pas de chemin de longueur } p \text{ de } x_i \text{ vers } x_j, \\ 1 & \text{s'il existe un chemin de longueur } p \text{ de } x_i \text{ vers } x_j, \end{cases} \quad (3)$$

La démonstration se fait par récurrence⁶.

⁶La propriété est vraie pour les chemins de longueurs 1 (les arcs et la matrice $M = M^{[1]}$), on suppose la propriété vraie jusqu'au rang p , en effectuant le produit de $M^{[p]} = m_{i,j}$ par $M = [a_{i,j}]$, si l'un des produits booléens $m_{i,k}a_{k,j}$ est égal à 1 et seulement dans ce cas, l'élément $m'_{i,j}$ de $M^{[p+1]} = [m'_{i,j}]$ vaut 1 ce qui correspond à l'existence d'un chemin de longueur p de x_i vers x_k et d'un arc de x_k vers x_j , donc bien d'un chemin de longueur $p + 1$ de x_i vers x_j .

3.3.3 Nombres de chemins.

Dans le cas d'un 1-graphe la matrice d'adjacence M d'un graphe donne le nombre de chemins de longueur 1 d'un sommet x_i vers un sommet x_j , (dans le cas d'un multigraphe, on pourrait remplacer la matrice M par une matrice dont les éléments $a_{i,j}$ auraient pour valeurs les nombres d'arcs de x_i à x_j).

Le produit $M \times M = M^2$, (qui n'est pas un produit booléen, mais le produit matriciel habituel), donne une matrice dont les éléments $m_{i,j}$ ont pour valeurs les nombres de chemins de longueur 2 de x_i à x_j . De même, d'une manière générale, les éléments $m_{i,j}$ de la matrice M^p ont pour valeurs les nombres de chemins de longueur p , de x_i à x_j .

Pour le graphe de la figure [fig. 24], on obtient les nombres de chemins de longueurs données en calculant les matrices suivantes :

$$\begin{array}{l}
 M = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad
 M^2 = \begin{bmatrix} 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 2 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad
 M^3 = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 2 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 \end{bmatrix} \\
 \\
 M^4 = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 2 & 1 \end{bmatrix} \quad
 M^5 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 2 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix} \quad
 M^6 = \begin{bmatrix} 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 2 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}
 \end{array}$$

3.3.4 Fermeture transitive d'un graphe.

Transitivité.

Définition 3.4 *Un graphe est transitif si chaque fois que l'on a à la fois un arc du sommet x vers le sommet y et un deuxième arc de y vers le sommet z , alors il existe un arc de x vers z .*

L'arc de x vers z associé aux deux arcs de x vers y et de y vers z , est parfois appelé un raccourci.

Propriété 3.5 *Si $G = (S, A)$ est un graphe, il existe au moins un graphe transitif $G' = (S, A')$ qui a même ensemble de sommets que G et qui est tel que $A \subset A'$.*

L'un de ces graphes transitif $G' = (S, A')$, contenu dans tous les autres, est le plus petit de ces graphes transitifs, on l'appelle fermeture transitive de G .

Un graphe G est transitif si et seulement si G est égal à sa fermeture transitive.

Construction de la fermeture transitive. L'idée est d'obtenir à partir de G un graphe transitif en ajoutant le moins possible d'arcs. La propriété affirme que c'est possible et qu'il existe un tel graphe transitif unique, contenu dans tous (donc le plus petit possible). Pour l'obtenir il suffit, pour chaque couple (x, y) de deux sommets, d'ajouter s'il le faut un arc de x vers y , lorsqu'il existe un chemin de x vers y dans G . Voir la figure [fig. 25].

En effectuant progressivement la somme $T = M \oplus M^{[2]} \oplus M^{[3]} \oplus M^{[4]} \oplus M^{[5]} \oplus \dots \oplus M^{[p]} \oplus \dots$ on peut observer qu'à partir d'une certaine valeur de p , la somme ne peut plus changer⁷. La matrice T obtenue est la matrice de la fermeture transitive du graphe.

L'élément $a_{i,j}$ de la matrice T vaut
$$\begin{cases} 0 & \text{s'il n'existe pas de chemin de } x_i \text{ vers } x_j, \\ 1 & \text{s'il existe un chemin de } x_i \text{ vers } x_j, \end{cases} \quad (4)$$

Supposons pour simplifier l'écriture que $T = M \oplus M^{[2]} \oplus M^{[3]} \oplus M^{[4]} \oplus M^{[5]}$, en utilisant le schéma de Hörner on peut écrire $T = (((M \oplus I) \otimes M \oplus I) \otimes M \oplus I) \otimes M \oplus I$, les additions à effectuer sont plus simples.

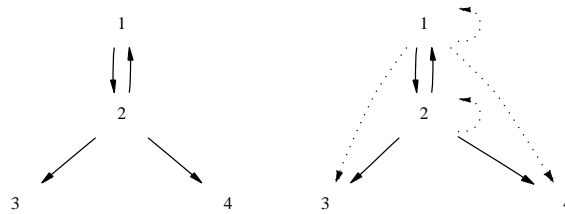


FIG. 25 – Graphe et sa fermeture transitive.

3.4 Valeur d'un chemin.

3.4.1 Carte routière.

Soit le graphe $G = (S, A)$, à chaque arc $a \in A$, associons un nombre, une valeur $v(a)$, v est une application de A dans \mathbb{R} ou dans un sous-ensemble de \mathbb{R} . À chaque chemin (a, b, \dots, e) du graphe on associe la somme des valeurs $v(a) + v(b) + \dots + v(e)$ de tous ses arcs.

Exemple : Soit S un ensemble de villes U, V, W, X, Y . A est l'ensemble des routes reliant certaines de ces villes. Il est habituel d'indiquer pour chaque route les distances en km entre les villes. À chaque itinéraire routier $U V W X Y$ de la ville U vers la ville Y on associe la distance totale de chacun des segments qui le composent.

3.4.2 Chemin de valeur minimale.

Propriété 3.6 Si a_1, a_2, \dots sont des arcs et si $a_1, a_2, \dots, a_i, \dots, a_j, \dots, a_p$ est un chemin de valeur minimale du sommet s vers le sommet s' , alors tout sous-chemin a_i, \dots, a_j est minimal entre les sommets x , origine de a_i et y , but de a_j .

L'explication est simple : si le sous-chemin a_i, \dots, a_j n'est pas minimal, alors en le remplaçant par un autre chemin $a_i, \dots, a_k, \dots, a_j$ de valeur moindre, on

⁷Comment pouvez-vous l'expliquer? Le nombre d'arcs est un ensemble fini.

obtient un chemin de plus petite valeur de s vers s' , ce qui contredit l'hypothèse et est absurde.

Voir la figure [fig. 26] et le chemin minimal de longueur $2 + 4 + 3 + 5$ de s vers s' .

Pour trouver le chemin minimal de s vers s' , on cherche les chemins minimaux de s vers tous les prédécesseurs u de s' , on ajoute aux valeurs de ces chemins la valeur de l'arc de u vers s' , il ne reste qu'à choisir le chemin correspondant à la plus petite valeur trouvée.

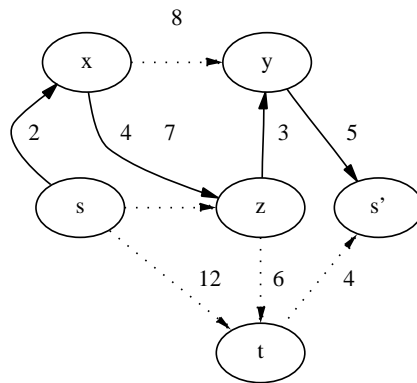


FIG. 26 – *Chemin minimal.*

3.4.3 Chemin de valeur maximale.

On peut se ramener au problème précédent en remplaçant la valeur d'un arc par le nombre opposé.

3.4.4 Problèmes classiques et difficiles.

Problème du voyageur de commerce.

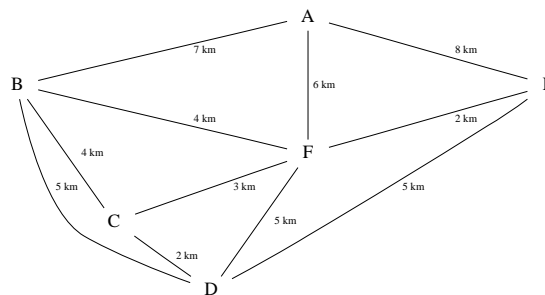


FIG. 27 – *Tournées du voyageur de commerce ou du postier chinois.*

Un voyageur de commerce doit visiter un certain nombre de villes et organiser un itinéraire le plus court possible.

En exemple prenons le graphe [fig. 27] qui a pour sommets les villes A, B, C, D, E, F , les arêtes du graphe représentent les routes, la valeur de chacune des arêtes est la distance entre deux villes reliées. Saurez-vous trouver le cycle le plus court passant une fois et une seule par tous les sommets ?

Problème du postier chinois. Problème posé pour la première fois par un chinois, Mei-ko Kwan, dont l'intention était d'optimiser la distribution du courrier dans une ville.

Le problème dit du *postier chinois* est de parcourir les rues d'une ville en passant au moins une fois dans chaque rue et de minimiser la distance totale parcourue. S'il existe un circuit eulérien le problème est résolu : tous les circuits eulériens ont même valeur, la somme des valeurs de tous les arcs, et c'est la plus petite valeur possible.

Si le graphe n'est pas eulérien, le circuit emprunte nécessairement plusieurs fois certains arcs.

Les postiers ne sont pas les seuls concernés, on peut trouver d'autres utilisations comme, par exemple, optimiser la distance parcourue par un robot nettoyeur de sols dans les couloirs d'un étage de bureaux, en un temps minimum.

Saurez-vous trouver le cycle le plus court passant au moins une fois par toutes les arêtes de la figure [fig. 27].

4 Niveaux.

Dans un graphe simple, orienté et sans circuits comme le graphe des diviseurs de la figure [fig. 1], on peut dessiner la figure de telle manière que toutes les flèches soient dirigées du haut vers le bas. De plus les sommets peuvent être placés sur différents niveaux 0, 1, 2, ... chacun des sommets étant au niveau le plus faible possible (et donc le plus haut possible sur la figure).

On peut voir que le niveau d'un sommet s est alors la longueur du plus long chemin d'origine une racine et d'extrémité s .

La figure [fig. 28] est le graphe de la relation « est un diviseur de » dont on n'a conservé que les arcs qui vont d'un niveau n au niveau $n + 1$ (c'est-à-dire juste au-dessous, sur la figure). Pour retrouver les arcs éliminés il suffit de prendre la fermeture transitive du graphe.

On peut remarquer que sur la figure [fig. 28], lorsqu'on décompose un naturel ainsi : $72 = 2^3 3^2$, (c'est l'écriture primaire de 72), le niveau de 72 dans l'arbre est la somme des deux exposants $3 + 2 = 5$.

Seul 1 est au niveau 0. Les nombres au niveau 1 sont $2 = 2^1, 3 = 3^1, 5, 7, \dots$, ce sont les nombres premiers.

5 Arbres et arborescences.

5.1 Arbres.

Définition 5.1 *Un arbre est un graphe connexe sans cycles. Une forêt est un graphe dont chaque composante connexe est un arbre : c'est un graphe sans cycles.*

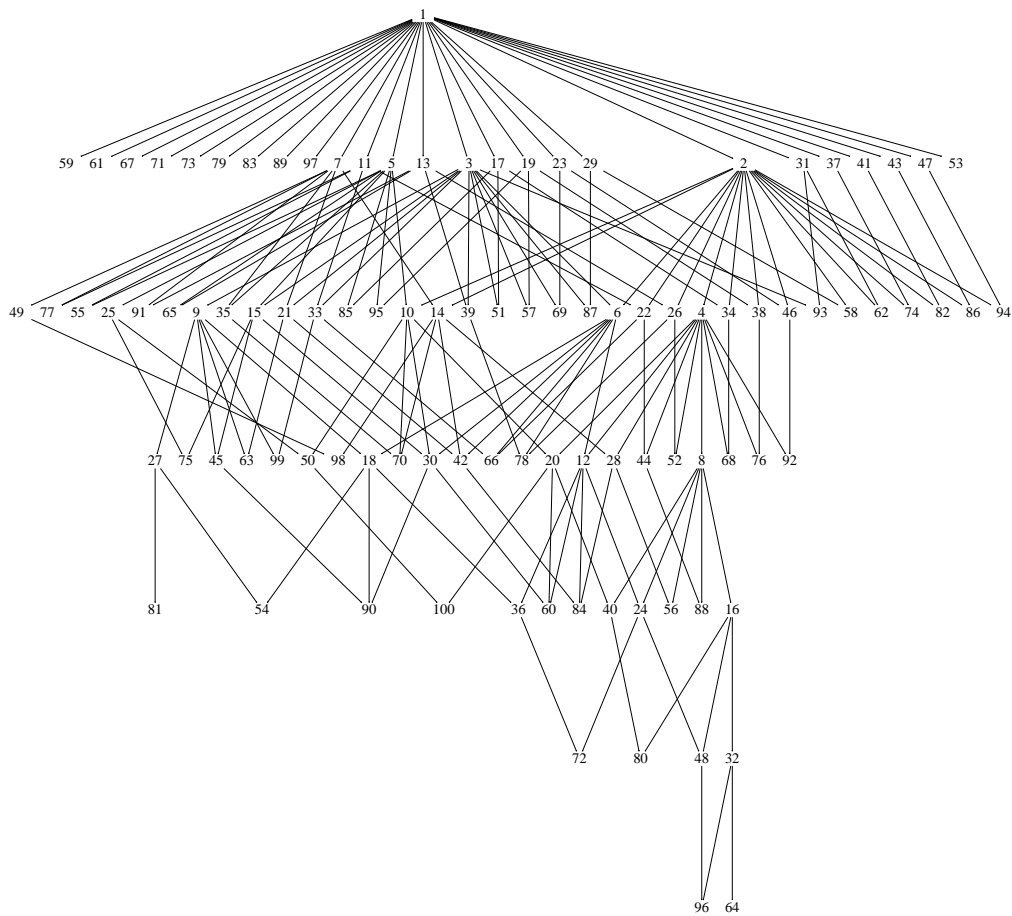


FIG. 28 – Niveaux.

Un arbre est un 1-graphe.

Voir la figure [fig. 29] pour l'un et l'autre cas orienté ou non orienté.

5.2 Arborescences.

Définition 5.2 Dans un graphe une racine est un sommet r tel que, pour tout sommet $x \neq r$, r est l'origine d'un chemin dont l'autre extrémité est x .

De la racine r on peut atteindre tout sommet en suivant un chemin du graphe.

Définition 5.3 Une arborescence est un arbre orienté muni d'une racine.

Voir la figure [fig. 30].

5.2.1 Arborescences binaires.

Une arborescence binaire Dans une arborescence binaire chaque sommet a soit deux successeurs et s'appelle un nœud, soit aucun successeur et s'appelle une feuille.

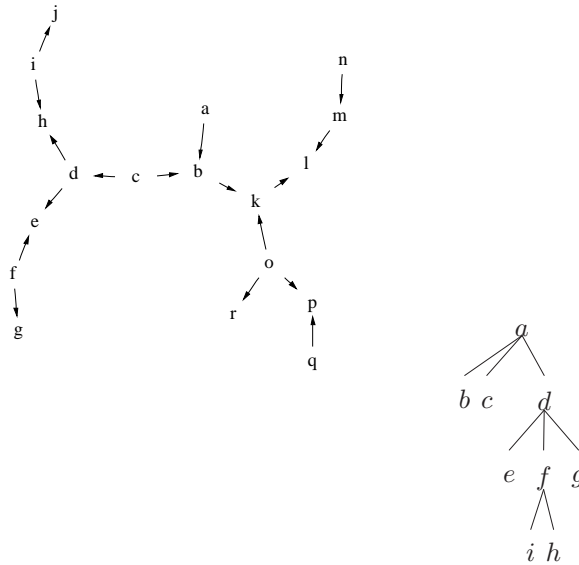


FIG. 29 – Arbres orienté et non-orienté.

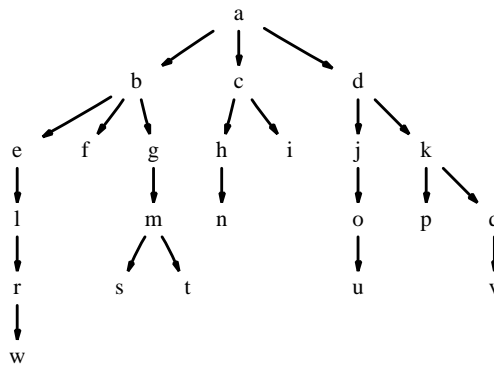


FIG. 30 – Arborescence.

6 Réseaux.

6.1 Réseau de transport.

La figure [fig. 31] représente un réseau de transport. Les valeurs indiquées sur les arcs sont les capacités, c'est-à-dire la quantité maximum (véhicules, marchandises, messages ...) pouvant utiliser l'arc.

Le problème du flot maximum est la recherche de la quantité maximum pouvant être acheminée de a vers j à travers le réseau, ou, ce qui revient au même, dans l'arc (j, a) (non tracé). En effet, la quantité arrivant en chaque sommet par les arcs entrants, doit être égale à la quantité en repartant par les arcs sortants, (tracer l'arc (j, a) permet aux sommets a et j de respecter cette règle).

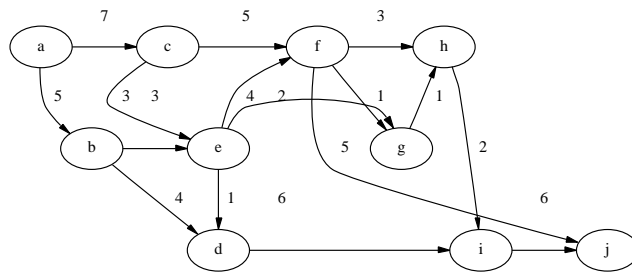


FIG. 31 – Réseau de transport.

7 Autres utilisations.

7.1 Probabilités et chaînes de Markov.

7.1.1 4 Faces consécutives au moins en n coups.

Le problème. Une expérience consiste à lancer n fois une pièce de monnaie. La variable aléatoire X prend la valeur 1 si on obtient au moins une fois une succession de 4 faces consécutives (ou plus de 4). X prend la valeur 0 dans le cas contraire.

Quelle est la loi de probabilité de X ?

Ce problème est différent du problème classique où la valeur de la variable aléatoire était le nombre d'apparitions total des faces. Il s'agissait d'un schéma de Bernoulli et de la loi binômiale.

Prenons un exemple, soit P P P F F P F P F F F P F F P F F F F P P une succession de lancers. Comptons le nombre de faces consécutives : 0 0 0 1 2 0 1 0 1 2 3 0 1 2 0 1 2 3 G G ... et écrivons 4 ou G = Gagné, dès que l'on observe 4 faces consécutives.



FIG. 32 – Jacob (Jacques) Bernoulli, 1654-1705, Suisse.

Le graphe correspondant est celui de la figure [fig. 33], les probabilités de passage des états 0, 1, 2, 3, 4 = G , les uns vers les autres sont inscrites en regard des arcs. Appelons $p_{i,j}$ la probabilité de passer de l'état i à l'état j . Seuls les arcs correspondant à des probabilités non nulles sont tracés. Ce graphe est appelé le *graphe des transitions*.

Au lieu de prendre la matrice booléenne du graphe, on peut écrire la matrice suivante, appelée *matrice de transition*,

$$M = [m_{i,j}] = \begin{bmatrix} 1/2 & 1/2 & 0 & 0 & 0 \\ 1/2 & 0 & 1/2 & 0 & 0 \\ 1/2 & 0 & 0 & 1/2 & 0 \\ 1/2 & 0 & 0 & 0 & 1/2 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

La première colonne et la première ligne correspondent à l'état 0, la dernière ligne et la dernière colonne à l'état G (ou 4).

En numérotant momentanément les lignes i et les colonnes j de 0 à 4, l'élément $m_{i+1,j+1} = p_{i,j}$ est la probabilité de passer de l'état i à l'état j .

La probabilité cherchée est

$$P_n = [1 \ 0 \ 0 \ 0 \ 0] M^n \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}.$$

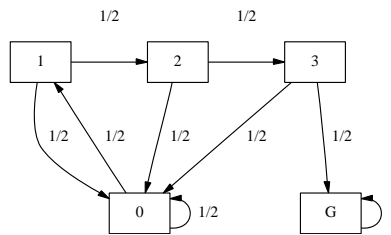


FIG. 33 – *Graphe des transitions (attente de 4 faces consécutives).*

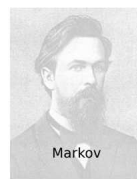


FIG. 34 – *Andrei Andreyevich Markov, 1856-1922, Russe (St. Pétersbourg).*

7.2 Problèmes d'affectation.

7.2.1 Couplage.

Binômes. Un groupe d'étudiants doit être réparti en binômes pour effectuer une certaine recherche. Pouvez-vous trouver une ou plusieurs répartitions qui respectent les préférences indiquées par le graphe de la figure [fig. 35]? (Chacun des étudiants appartient à un binôme unique, chaque binôme est composé de deux étudiants, c'est une paire).

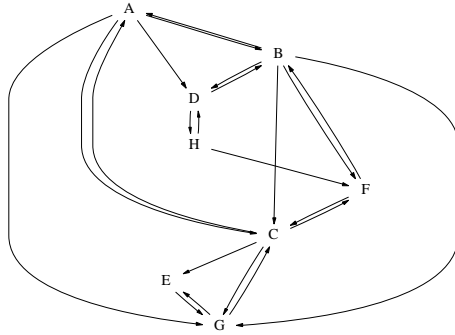


FIG. 35 – Binômes d'étudiants.

Affecter 5 machines aux 5 ouvriers. Comment affecter le plus grand nombre possible d'ouvriers sur des machines qu'ils savent utiliser? Les n ouvriers sont O_1, \dots, O_n et les m machines M_1, \dots, M_m . Les compétences des ouvriers à conduire les machines sont indiquées sur le graphe de la figure [fig. 36], on remarque qu'il s'agit d'un graphe biparti. La matrice d'incidence est

$$A = \begin{bmatrix} (1) & 1 & 0 & 0 & 0 \\ 1 & 1 & (1) & 1 & 0 \\ 0 & (1) & 1 & 0 & 0 \\ 0 & 0 & 1 & (1) & 1 \\ 0 & 0 & 0 & 1 & (1) \end{bmatrix}.$$

Pour trouver le couplage indiqué par la mise entre parenthèses de certains (1) de la matrice A , on procède ainsi :

- choisir l'une des lignes comportant le nombre minimal de 1, encadrer l'un de ces (1),
- effacer la ligne et la colonne de ce (1) encadré,
- recommencer tant qu'il reste encore un 1 non rayé.

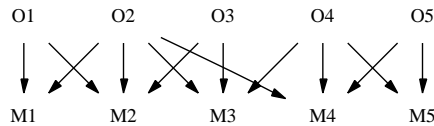


FIG. 36 – Graphe des compétences des ouvriers.

7.3 Coloration des sommets d'un graphe.

Le *nombre chromatique* d'un graphe est le plus petit nombre de couleurs nécessaires pour colorier les sommets, de sorte que deux sommets adjacents distincts ne soient pas de la même couleur.

Le problème des examens écrits. Soit Y un ensemble d'étudiants et X l'ensemble de toutes les épreuves possibles. On désire que tous les étudiants devant passer l'écrit x le fassent évidemment ensemble. Chaque étudiant ne passe qu'une épreuve par jour. Quel est le nombre minimum de jours nécessaires pour tous les étudiants passent leurs examens ?

En prenant X comme ensemble de sommets et en traçant une arête entre les sommets x et x' lorsqu'un étudiant au moins se présente aux deux épreuves x et x' , le problème revient à chercher le nombre chromatique du graphe.

Le problème des écolières. Les 15 jeunes filles d'un pensionnat sortent tous les jours en promenade⁸ par rangées de trois. Est-il possible de les promener pendant 7 jours consécutifs sans que deux jeunes filles ne soient ensemble (dans un groupe de trois) plus d'une fois ?

Ce problème a été résolu par Cayley⁹

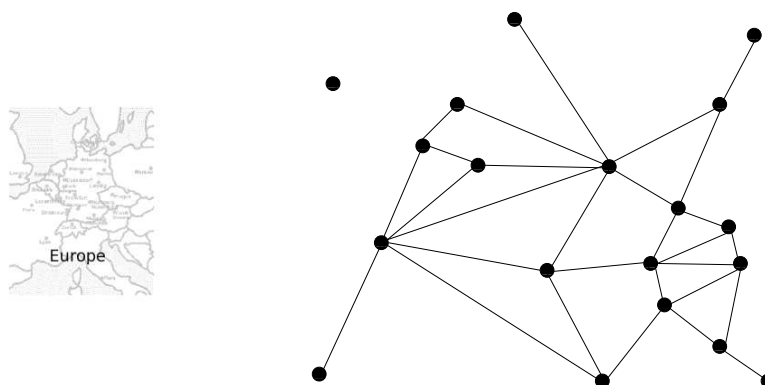


FIG. 37 – Carte d'Europe.

7.3.1 Problème des 4 couleurs.

Est-il possible de colorier les pays de n'importe quelle carte de géographie en 4 couleurs ou moins ? (Deux pays frontaliers doivent être de couleurs différentes et chaque pays n'est formé que d'une seule région).

On transforme la carte de géographie en un graphe planaire dont les sommets sont les pays et les arcs relient deux pays ayant une frontière commune.

Il semble que tout graphe planaire soit coloriable en seulement 4 couleurs, les

⁸Ce problème date un peu !

⁹Artur Cayley, né le 16 Août 1821 à Richmond, Surrey, Angleterre. Décédé le 26 Janvier 1895 à Cambridge, Angleterre

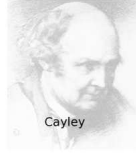


FIG. 38 – Arthur Cayley, 1821-1895, anglais .

deux seules démonstrations à ce jour utilisent des calculs longs et complexes sur ordinateur et ne sont pas vérifiables *à la main*. Voir en particulier [22].

7.4 Noyaux.

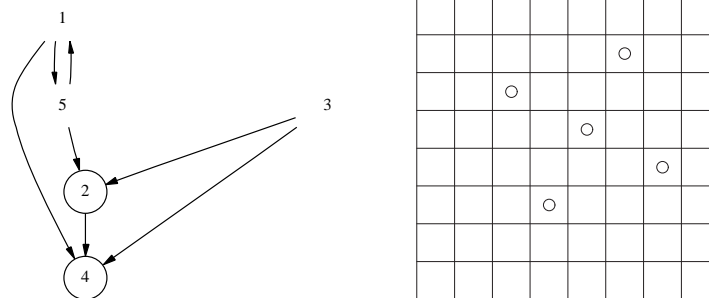


FIG. 39 – Radars. Dames

7.4.1 Nombre d'absorption.

Un sous-ensemble $A \subset S$ de sommets est dit *absorbant* ou *stable extérieurement* si tout sommet s qui n'est pas dans A , possède au moins un successeur dans A .

Le *nombre d'absorption* du graphe est le nombre d'éléments du plus petit ensemble absorbant A .

Le problème des radars. S est un ensemble de régions contenant chacune un radar. Si une région s peut être surveillée par le radar de la région s' , on trace un arc de s' vers s . Quel est le nombre minimum de radars nécessaires pour contrôler toutes les régions? Voir la figure [fig. 39].

le problème des 5 dames. Cinq dames suffisent pour contrôler toutes les 8×8 cases d'un échiquier, voir la figure [fig. 39].

7.4.2 Noyaux.

Le sous-ensemble de sommets $A \subset S$ est *stable* si tout sommet $s \in A$ n'a aucun successeur dans A .

Le sous-ensemble de sommets $A \subset S$ est un *noyau* s'il est stable et absorbant.

Jeu. Le premier joueur énonce un nombre entier entre 10 et 20. À tour de rôle les joueurs choisissent un nombre égal au précédent diminué de 1, 2 ou 3. Le premier qui arrive à 0 est le gagnant.

Le graphe a pour sommets $0, 1, \dots, 20$ et ses arcs relient les entiers comme il a été dit plus haut, par exemple 17 à 16, 17 à 15 et 17 à 14.

Les positions gagnantes sont les sommets d'un noyau contenant 0. Ce noyau est $A = \{0, 4, 8, 12, 16, 20\}$.

A est bien un noyau, il est stable car les éléments de $A = \{0, 4, 8, 12, 16, 20\}$ n'ont pas de successeur dans A , (les éléments de A ne sont pas adjacents). A est absorbant car tout sommet qui n'est pas dans A , possède un successeur dans A , (par exemple 15 qui n'est pas dans A , a pour successeur 12 qui est dans A).

7.5 Ordonnancement de tâches.

7.5.1 Recette de crêpes.

Le sigle PERT signifie *Program Evaluation and Review Technic*, que l'on traduit par « Méthode de planification à base de recherche opérationnelle, ou technique d'établissement et de remise à jour des programmes. »

Autre sigle : WBS *Work Breakdown Structure*, « Méthode de gestion de projet sous forme d'organigramme. »

| tâche | durée en s | étape | tâche à exécuter | tâches préliminaires |
|--|------------|-------|------------------|----------------------|
| Mettre toute la farine dans un récipient | 3 | A | A | |
| Casser les oeufs sur la farine | 30 | B | B | A |
| Verser lentement le lait tout en remuant | 600 | C | C | A, B |
| Verser de la pâte dans la poêle | 3 | D | D | C, G |
| Couper les bananes en rondelles | 300 | E | E | |
| Mélanger les rondelles de bananes et le rhum | 30 | F | F | E |
| Chauffer la poêle | 120 | G | G | |
| Flamber | 10 | H | H | J |
| Cuire une crêpe | 10 | I | I | D |
| Verser les rondelles de bananes sur la crêpe | 10 | J | J | F, I |
| Goûter la première crêpe | | K | K | J |

FIG. 40 – Recette des crêpes flambées.

Les tableaux des tâches. Voir la figure [fig. 40], le premier tableau donne les durées de toutes les tâches à accomplir (faire la pâte ...) avant même de déguster¹⁰ la première crêpe.

¹⁰La recette des crêpes flambées aux bananes est indiquée sans garantie d'aucune sorte, n'étant pas cuisinier et n'ayant jamais mangé de crêpe aux bananes, je n'ai pas vérifié la

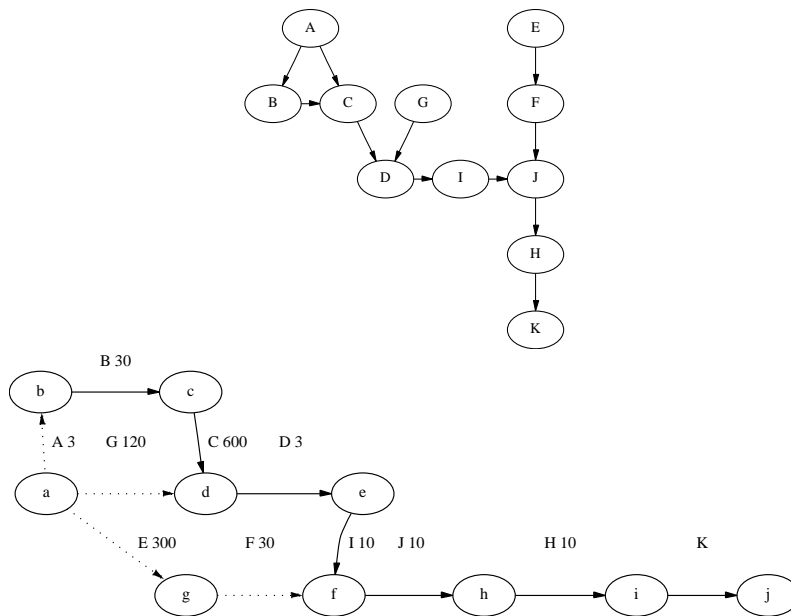


FIG. 41 – Préparation des crêpes.

Les graphes. Le premier graphe de la figure [fig. 41] a pour sommets les différentes tâches A, B, ..., K, il est construit à partir du tableau des précédences de la figure précédente [fig. 40].

Le deuxième graphe de la figure [fig. 41] est le *graphe d'ordonnancement* il a pour arcs les tâches, ses sommets a, b, \dots, j correspondent aux prédécesseurs et successeurs des arcs. Les durées en secondes des tâches sont indiquées sur le schéma. Le sommet a est une racine du graphe.

Le chemin $abcdefhij$ est appelé *chemin critique*, sa valeur est la durée minimale nécessaire pour que toutes les opérations puissent être effectuées.

Il reste à déterminer les dates (instants) *au plus tôt* et les dates *ou plus tard* de début de chacune des tâches, ce qui ne devrait pas être difficile, si l'on veut que l'opération totale ne prenne pas plus de temps que la durée minimale calculée.

7.5.2 Ordonnement de tâches, activités sur les sommets.

L'exemple (données, traitement et solution) a été réalisé automatiquement¹¹ par des moyens informatiques et ne correspond pas à un problème réel.

Données et résultats. Le tableau [fig. 42] indique les noms des tâches à accomplir dans la colonne 2. Pour chacune des tâches t , la durée est écrite dans la colonne 3 et les noms des tâches qui doivent être terminées avant que la tâche t ne commence sont indiqués dans la colonne 4.

Les valeurs des colonnes suivantes sont calculées à partir des données précédentes.

Lorsque la marge est nulle, la tâche est critique, retarder les dates de son exécution serait retarder la durée de tout le projet.

Si toutes les tâches sont exécutées entre leurs dates de début au plus tôt et de fin au plus tard, la durée du projet est de 259 unités de temps.

Graphe. La figure [fig. 43] donne le graphe de la relation de précédence entre les différentes tâches.

Les 14 niveaux indiqués dans le tableau [fig. 42] se retrouvent sur le graphe, ils vont du niveau 0, en haut de la figure, au plus grand niveau 13, en bas. Les tâches critiques, dont les noms sont cerclés, sont disposées du niveau 0 au niveau 13, (mais qui pourrait être inférieur au niveau maximum dans un autre exemple). Les arcs ne reliant pas deux tâches critiques sont dessinés en pointillés.

Diagramme de Gantt. La figure [fig. 44] donne le diagramme de Gantt. Pour chacune des tâches, le segment indique les dates de début au plus tôt et de fin au plus tôt, la marque + indique la date de fin au plus tard. Lorsque la tâche est critique, la marque + est à l'extrémité supérieure du segment. La valeur de la marge est le laps de temps compris entre l'extrémité supérieure du segment et le signe +.

¹¹Le programme principal d'environ 300 lignes (génération aléatoire des données, calcul des niveaux, des tâches critiques, des dates et des marges) est écrit en AWK. Ce programme permettrait le traitement de problèmes plus importants. (La durée de calcul varie comme le cube du nombre de tâches et un langage interprété comme Awk conduit à des temps très longs lorsque le nombre de sommets du graphe dépasse plusieurs centaines. Les 30 sommets de l'exemple se calculent en une fraction de seconde, pour 3 000 sommets, la durée est multipliée par 1 Million et est probablement de plusieurs heures).

Les fichiers et commandes produits par le programme principal sont

- un fichier dans le langage de DOT pour l'obtention du diagramme sagittal du graphe,
- un fichier de données pour le diagramme de Gantt et une commande de traitement par GNUPLOT,
- un fichier L^AT_EX qui donne le document final.

Citons parmi les outils mathématiques et les algorithmes utilisés, la recherche de la fermeture transitive du graphe avant l'élimination de tous les raccourcis qui ne feraient que rendre la figure moins lisible.

| n | Tâche t_n | Durée d_n | Précédentes | Suivantes | Niveau r_n | Marge m_n | Début au plus tôt | Fin au plus tôt | Début au plus tard | Fin au plus tard |
|-----|----------------|----------------|----------------------------------|----------------------------------|-----------------|----------------|-------------------|-----------------|--------------------|------------------|
| 1 | t_1 | 4 | | t_2 | 0 | 0 | 0 | 4 | 0 | 4 |
| 2 | t_2 | 5 | t_1 | t_3, t_7 | 1 | 0 | 4 | 9 | 4 | 9 |
| 3 | t_3 | 27 | t_2 | t_4, t_{11} | 2 | 0 | 9 | 36 | 9 | 36 |
| 4 | t_4 | 29 | t_3 | t_5 | 3 | 0 | 36 | 65 | 36 | 65 |
| 5 | t_5 | 24 | t_4 | t_6 | 4 | 0 | 65 | 89 | 65 | 89 |
| 6 | t_6 | 19 | t_5 | t_8 | 5 | 0 | 89 | 108 | 89 | 108 |
| 7 | t_7 | 11 | t_2 | t_8, t_{29} | 2 | 88 | 9 | 20 | 97 | 108 |
| 8 | t_8 | 2 | t_6, t_7 | t_9, t_{15}, t_{27} | 6 | 0 | 108 | 110 | 108 | 110 |
| 9 | t_9 | 23 | t_8 | t_{10} | 7 | 0 | 110 | 133 | 110 | 133 |
| 10 | t_{10} | 4 | t_9 | t_{12} | 8 | 0 | 133 | 137 | 133 | 137 |
| 11 | t_{11} | 15 | t_3 | t_{13}, t_{15} | 3 | 125 | 36 | 51 | 161 | 176 |
| 12 | t_{12} | 39 | t_{10} | $t_{13}, t_{14}, t_{16}, t_{17}$ | 9 | 0 | 137 | 176 | 137 | 176 |
| 13 | t_{13} | 36 | t_{11}, t_{12} | t_{21}, t_{22}, t_{26} | 10 | 0 | 176 | 212 | 176 | 212 |
| 14 | t_{14} | 2 | t_{12} | t_{18}, t_{19} | 10 | 16 | 176 | 178 | 192 | 194 |
| 15 | t_{15} | 8 | t_8, t_{11} | t_{18} | 7 | 76 | 110 | 118 | 186 | 194 |
| 16 | t_{16} | 9 | t_{12} | t_{21} | 10 | 27 | 176 | 185 | 203 | 212 |
| 17 | t_{17} | 12 | t_{12} | t_{18} | 10 | 6 | 176 | 188 | 182 | 194 |
| 18 | t_{18} | 34 | t_{14}, t_{15}, t_{17} | t_{23}, t_{24} | 11 | 6 | 188 | 222 | 194 | 228 |
| 19 | t_{19} | 8 | t_{14} | t_{21}, t_{24}, t_{28} | 11 | 26 | 178 | 186 | 204 | 212 |
| 20 | t_{20} | 6 | | t_{21}, t_{28} | 0 | 206 | 0 | 6 | 206 | 212 |
| 21 | t_{21} | 25 | $t_{13}, t_{16}, t_{19}, t_{20}$ | t_{25} | 12 | 0 | 212 | 237 | 212 | 237 |
| 22 | t_{22} | 5 | t_{13} | t_{25}, t_{28} | 11 | 20 | 212 | 217 | 232 | 237 |
| 23 | t_{23} | 9 | t_{18} | t_{25}, t_{28} | 12 | 6 | 222 | 231 | 228 | 237 |
| 24 | t_{24} | 15 | t_{18}, t_{19} | t_{26} | 12 | 11 | 222 | 237 | 233 | 248 |
| 25 | t_{25} | 22 | t_{21}, t_{22}, t_{23} | | 13 | 0 | 237 | 259 | 237 | 259 |
| 26 | t_{26} | 11 | t_{13}, t_{24} | | 13 | 11 | 237 | 248 | 248 | 259 |
| 27 | t_{27} | 25 | t_8 | | 7 | 124 | 110 | 135 | 234 | 259 |
| 28 | t_{28} | 13 | $t_{19}, t_{20}, t_{22}, t_{23}$ | | 13 | 15 | 231 | 244 | 246 | 259 |
| 29 | t_{29} | 19 | t_7 | | 3 | 220 | 20 | 39 | 240 | 259 |

FIG. 42 – Tableau des tâches

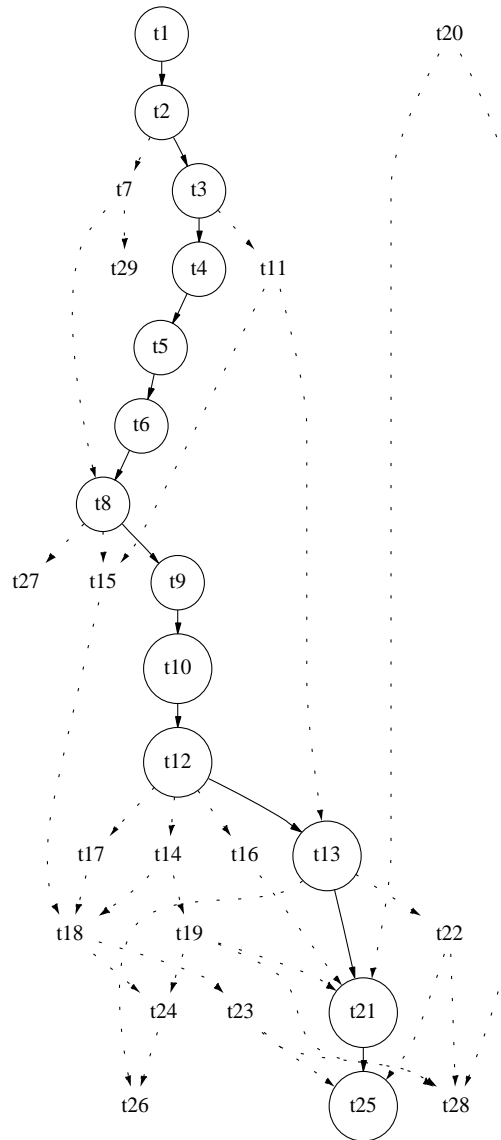


FIG. 43 – Diagramme des tâches.

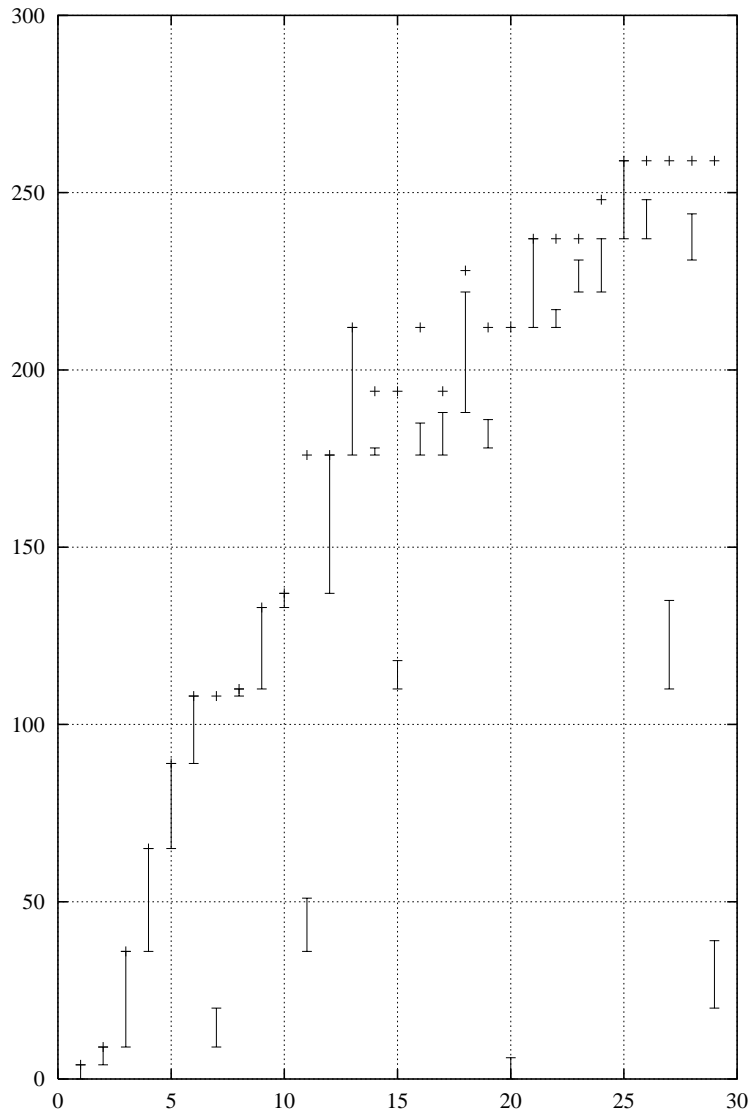


FIG. 44 – Diagramme de Gantt.

A Les graphes et le programme de BTS Informatique de gestion.

A.1 Le programme.

GRAPHES

Cette initiation aux graphes orientés doit être menée en étroite concertation avec les enseignements de l'informatique et de la gestion où cette étude est poursuivie.

L'objectif est d'introduire et de mettre en oeuvre, dans des situations concrètes très élémentaires et sans théorie générale, des algorithmes permettant de résoudre les problèmes figurant dans la rubrique de travaux pratiques.

Modes de représentation d'un graphe orienté : représentation géométrique, tableau des successeurs ou des prédécesseurs, matrice adjacente (booléenne). Chemin, circuit, boucle, chemin hamiltonien.

Arborescence.

Longueur d'un chemin, chemin optimal.

La définition d'un graphe orienté n'est pas au programme.

La notion de connexité étant hors programme, on se limitera à la présentation d'exemples simples d'arborescences à partir de leur représentation géométrique, sans recherche d'une caractérisation générale.

On observera l'importance du résultat : tout sous chemin d'un chemin optimal est optimal.

Travaux pratiques

Exemples de mise en oeuvre d'algorithmes permettant d'obtenir pour un graphe

- les chemins de longueur p ,
- la fermeture transitive,
- les niveaux, dans le cas d'un graphe sans circuit,
- les chemins de valeur minimale (ou le cas échéant de valeur maximale).

Exemple de résolution de problèmes d'ordonnancement par la méthode des potentiels ou la méthode PERT.

A partir d'exemples très élémentaires et sans introduire une théorie générale, on montrera l'intérêt des méthodes matricielles mettant en oeuvre l'addition et la multiplication booléennes des matrices adjacentes. Dans une évaluation en mathématiques, tout énoncé relatif à ces algorithmes doit comporter des indications sur la méthode à suivre.

Il s'agit d'un premier contact avec des méthodes largement utilisées en gestion ; ces méthodes ne peuvent faire l'objet d'aucune évaluation en mathématiques.

A.2 Domaines d'utilisation.

- élaboration, gestion des réseaux,
- élaboration de circuits (VLSI),
- problèmes généraux de routage, recherches de plus courts chemins,
- ordonnancement des tâches, en particulier dans les systèmes parallèles,
- structures de données.

A.3 Remarques

Une connaissance précise d'une partie des définitions et propriétés ci-dessous n'est pas au programme mais il est quand même nécessaire de savoir ce qu'est un graphe, une arborescence, un chemin ...

Le but des quelques pages qui suivent est de permettre

- i. de préciser l'une ou l'autre des notions abordées, le plus souvent simples si on leur consacre quelques minutes ; par exemple un graphe c'est « des sommets, des arcs et savoir dire l'origine et le but de chacun des arcs » ; la définition n'est que la transcription en termes plus mathématiques de cette explication, sans rien ajouter de plus ;
- ii. de répondre à des interrogations telles « quelle différence y a-t-il entre une arborescence et un arbre ? » ;
- iii. de lire le programme : le mot *connexité* est écrit dans le programme, la notion de connexité n'est pas à apprendre, (il suffira de sauter les paragraphes qui ne sont manifestement pas au programme, de toute façon la quasi totalité du temps consacré à l'étude des graphes sera celui passé à résoudre les exercices) ;
- iv. de reprendre des petits problèmes que vous avez peut-être essayé de résoudre un jour sans avoir trouvé de solution (les trois villas par exemple) ;
- v. d'introduire des figures correspondant à des situations ou à des graphes les plus variés possibles.

Index

- 1–graphe, 1
- absorbant (sous-ensemble), 27
- absorption (nombre d'), 27
- affectation, 24
- algorithmique., 3
- application, 5, 6
- arborescence, 21
- arborescence, 21
- arborescence binaire, 21
- arbre, 3
- arc, 1

- Bernoulli, 23
- binôme, 24
- binômiale (loi), 23
- biparti (graphe), 2
- boucle, 1, 6
- but, 5, 6

- capacité, 22
- carte de géographie, 26
- casse-tête, 11
- cavalier, 11
- chaîne, 5
- chaîne de Markov, 23
- chaîne eulérienne, 12
- chemin, 5
- chemin élémentaire, 5
- chemin critique, 29
- chemin simple, 5
- chemin hamiltonien, 10
- chemins (nombre de), 17
- chromatique (nombre), 26
- circuit, 4, 6, 20
- circuit (sans), 20
- circuit eulérien, 14
- code d'accès, 14
- Collatz, 4
- coloration, 26
- composante connexe, 6
- conjecture, 4
- connexe, 2, 6
- connexe (composante), 6
- couleur, 26
- couplage, 24
- couple, 1
- court (le plus), 14, 19, 20, 34
- critique (chemin), 29
- critiques (tâches), 30
- cycle, 5, 20
- cycle eulérien, 13

- diagramme de Gantt, 30
- Debruijn, 14
- degré, 15
- degré d'un sommet, 2
- degré entrant d^- , 2
- degré sortant d^+ , 2
- diagramme sagittal, 7
- diagramme des tâches, 32
- disjoint, 2
- diviseur, 1, 20
- division, 1

- échiquier, 11
- écriture primaire, 20
- état final, 10
- état initial, 10
- Euler, 2, 10, 11
- eulérien, 20
- eulérien (circuit), 14
- eulérienne (chaîne), 12
- expérience, 23
- extrémité, 1, 2
- extrémité initiale, 5
- extrémité terminale, 1, 5

- fermeture transitive, 20
- fermeture transitive, 30
- fermeture transitive d'un graphe, 17
- feuille, 21
- flèche, 7
- flot, 22
- flot maximum (problème du), 22

- Gantt, 30
- gestion, 28
- graphe, 1
- 1–graphe orienté, 1
- 3–graphe, 4
- p –graphe, 5
- 1–graphe, 5
- graphe biparti, 2, 25
- graphe d'ordonnancement, 29
- graphe des transitions, 23
- graphe non orienté, 6
- graphe orienté, 5, 20
- graphe planaire, 2
- graphe simple, 1, 20
- graphe transitif, 17

- Hamilton, 10
- hamiltonien (chemin), 10
- Hörner, 18

- impair, 2
- incidence (matrice), 25
- informatique, 4
- initiale (extrémité), 1
- interdépendance, 4
- itinéraire, 19

- labyrinthe, 10
- liste, 5
- liste ordonnée, 5
- loi de probabilité, 23
- loi binômiale, 23
- longueur d'un chemin, 16

- marge, 30
- Markov, 23
- matrice d'incidence, 25
- matrice de transition, 24
- maximum, 22
- minimiser, 20
- minimum, 20
- mot, 14
- multigraphe, 4, 5

- niveau, 30
- niveaux, 20
- nœud, 21
- nombre de chemins, 17
- nombre premier, 20
- nombre d'absorption, 27
- nombre chromatique, 26
- non orienté (graphe), 6
- noyau d'un graphe, 27, 28

- opérationnelle (recherche), 28
- optimiser, 20
- ordonnancement (graphe d'), 29
- ordre du graphe, 2
- organigramme, 28
- orientation, 5, 6
- orienté (graphe), 20
- origine, 5, 6

- pair, 2, 3, 15
- paire, 6, 25
- passage (probabilités de), 23
- PERT, 28
- Petersen, 12
- p -graphe, 5
- planaire (graphe), 2, 26
- planification, 28
- postier chinois (problème du), 20
- prédécesseur, 2, 5
- préférences, 25
- primaire (écriture), 20

- probabilité, 23
- probabilité (loi de), 23
- probabilités de passage, 23
- problème du chemin, 10
- problème du flot maximum, 22
- problème des 4 couleurs, 26
- problème des 5 dames, 27
- problème des écolières, 26
- problème des examens écrits, 26
- problème du postier chinois, 20
- problème du voyageur de commerce, 19
- programme, 11
- projet, 28
- pseudo-circuit, 6
- pseudo-cycle, 6

- quadruplet, 5

- raccourci, 17, 30
- racine, 21
- recherche opérationnelle, 28
- relation d'Euler, 2
- répartition, 25
- réseau, 22
- réseau de transport, 22

- sagittal (diagramme), 7
- schéma, 7
- schéma de Bernoulli, 23
- schéma de Hörner, 18
- simple (graphe), 1, 20
- sommet, 1
- sommet (degré d'un), 2
- sous-ensemble, 1
- sous-ensemble absorbant, 27
- stable (sous-ensemble), 28
- stable extérieurement (sous-ensemble), 27
- successeur, 2, 5
- suite de Debruijn, 14

- table, 7
- tableau, 7
- tâche, 28, 34
- tâches critiques, 30
- terminale (extrémité), 1
- transitive (fermeture transitive d'un graphe), 17
- transitive (fermeture transitive du graphe), 30
- transitivité, 17

- valeur, 20
- valeur d'un chemin, 18
- variable aléatoire, 23
- villas (problème des 3), 2

voyageur de commerce (problème du), 19

WBS, 28

Références

- [1] Claude Berge. *Graphes et hypergraphes*. DUNOD, Paris, 1973.
- [2] Paul E. Black. *Dictionary of Algorithms, Data Structures, and Problems*. <http://www.nist.gov/dads/>.
- [3] Alexander Bogomolny. *Wythoff's Nim*. <http://www.cut-the-knot.com/pythagoras/wythoff.html>.
- [4] Encyclopædia Britannica. Britannica.com. <http://search.britannica.com/search?query=euler>.
- [5] Chris K. Caldwell. *Graph Theory Tutorials*, 1995. <http://www.utm.edu/departments/math/graph/>.
- [6] Michel Carvallo. *Monographie des treillis et Algèbre de Boole*. gauthier-villars, Paris, 1966.
- [7] David Applegate Robert Bixby Vasek Chvatal William Cook. *Solving traveling salesman problem*. <http://www.keck.caam.rice.edu/tsp/>.
- [8] Pierluigi Crescenzi and Viggo Kann. A compendium of NP optimization problems. <http://www.nada.kth.se/~viggo/wwwcompendium/wwwcompendium.html>.
- [9] Joseph Culberson. *Graph Coloring Page*. <http://web.cs.ualberta.ca/~joe/Coloring/index.html>.
- [10] Reinhard Diestel. *Graph Theory*. Springer, New York, 2000. <http://www.math.uni-hamburg.de/home/diestel/index.html>.
- [11] David Eppstein. *Design and Analysis of Algorithms*. <http://www.ics.uci.edu/~eppstein/161/syl.html>.
- [12] Dot and Dotty (Graphviz). <http://seclab.cs.ucdavis.edu/~hoagland/Dot.html>.
- [13] André Arnold Irène Guessarian. *Mathématiques pour l'informatique*. MASSON, 120, Bd. St-Germain, 75280 Paris Cedex 06, 1997.
- [14] A. Kaufmann A. Henry-Labordère. *Méthodes et modèles de la recherche opérationnelle*. DUNOD, Paris, 1973.
- [15] Equipe Graphes Laboratoire Leibniz. *Un aperçu de la théorie des graphes*. <http://www-leibniz.imag.fr/GRAPH/francais/apercu.html>.
- [16] Jeffrey C. Lagarias. *The $3x + 1$ Problem and its Generalisations*. *American Mathematical Monthly Volume 92, 1985, 3 - 23.*, 1985. <http://www.cecm.sfu.ca/organics/papers/lagarias/vault/paper.ps>.
- [17] Stephen C. Locke. *Graph Theory*. <http://www.math.fau.edu/locke/graphthe.htm>.
- [18] Daniel E. Loeb. *Gagner au Jeu de Nim*. LaBRI, 1995. <http://dept-info.labri.u-bordeaux.fr/~loeb/nim/>.
- [19] Christopher P. Mawata. *Graph Theory Lessons*. <http://www.utc.edu/~cpmawata/petersen/>.
- [20] John M. Harris Jeffrey L. Hirst Michael J. Mossinghoff. *Combinatorics and Graph Theory*. Springer, New York, 2000.
- [21] Didier Müller. *Lexique sur les graphes*. <http://www.jura.ch/lcp/cours/dm/graphes/lexique.html>.
- [22] Paul Seymour Neil Robertson, Daniel P. Sanders and Robin Thomas. *The Four Color Theorem*. <http://www.math.gatech.edu/~thomas/FC/fourcolor.html>.
- [23] University of St Andrews Scotland. The MacTutor History of Mathematics archive. <http://www-groups.dcs.st-and.ac.uk/~history/index.html>.
- [24] J. C. Derniame C. Pair. *Problèmes de cheminement dans les graphes*. DUNOD, Paris, 1971.
- [25] *Programmes BTS Informatique de gestion*. <http://www.ac-rennes.fr/pedagogie/maths/btsv7/prog/proginfg.htm>.

- [26] Gordon F. Royle. *Counting Set Covers and Split Graphs*. *Journal of Integer Sequences*, 3, 2000.
<http://www.research.att.com/~njas/sequences/JIS/>.
- [27] Daniel P. Sanders. *The Graph Theory White Pages*.
<http://www.cs.columbia.edu/~sanders/>.
- [28] Robert A. Sulanke. *Moments of Generalized Motzkin Paths*. *Journal of Integer Sequences*, 3, 2000.
<http://www.research.att.com/~njas/sequences/JIS/>.
- [29] Theseus. *Bibliography of Julius Petersen*.
http://www.math.ku.dk/thspetersen_j/.
- [30] Jonathan Gross Jay Yellen. *Graph Theory and ITS APPLICATIONS*.
<http://www.graphtheory.com/index.htm>.

Table des figures

| | | |
|----|---|----|
| 1 | <i>Graphe de la relation « est diviseur de »</i> | 1 |
| 2 | <i>Les trois villas</i> | 2 |
| 3 | <i>Calcul de $(ab + cd)/e$</i> | 3 |
| 4 | <i>Graphes et algorithmes</i> | 3 |
| 5 | <i>Interdépendances des chapitres. [13]</i> | 4 |
| 6 | <i>Arcs multiples</i> | 5 |
| 7 | <i>Cycle, circuit, pseudo-cycle</i> | 6 |
| 8 | <i>Graphe non-connexe</i> | 6 |
| 9 | <i>Graphe non-orienté</i> | 7 |
| 10 | <i>Diagramme cartésien, tableaux, diagramme sagittal</i> | 8 |
| 11 | <i>Matrice d'adjacence</i> | 8 |
| 12 | <i>Chèvre, loup, chou et passeur : C L X P</i> | 10 |
| 13 | <i>Labyrinthe et son graphe</i> | 11 |
| 14 | <i>Voyage autour du monde (Hamilton)</i> | 11 |
| 15 | <i>Course du cavalier (quelques solutions)</i> | 12 |
| 16 | <i>Course du cavalier (programme)</i> | 12 |
| 17 | <i>Trois solutions du casse-tête</i> | 13 |
| 18 | <i>Graphe de Petersen. Julius Peter Christian Petersen, 1839-1910, Danemark</i> | 13 |
| 19 | <i>Gottfried Wilhelm von Leibniz, 1646-1716, Allemagne</i> | 13 |
| 20 | <i>Cycles eulériens ? (Euler)</i> | 14 |
| 21 | <i>Kœnigsberg</i> | 14 |
| 22 | <i>La Pregel et l'île de Kneiphof à Kœnigsberg</i> | 15 |
| 23 | <i>Codes d'accès</i> | 15 |
| 24 | <i>Longueurs de chemins</i> | 16 |
| 25 | <i>Graphe et sa fermeture transitive</i> | 18 |
| 26 | <i>Chemin minimal</i> | 19 |
| 27 | <i>Tournées du voyageur de commerce ou du postier chinois</i> | 19 |
| 28 | <i>Niveaux</i> | 21 |
| 29 | <i>Arbres orienté et non-orienté</i> | 22 |
| 30 | <i>Arborescence</i> | 22 |
| 31 | <i>Réseau de transport</i> | 23 |
| 32 | <i>Jacob (Jacques) Bernoulli, 1654-1705, Suisse</i> | 23 |
| 33 | <i>Graphe des transitions (attente de 4 faces consécutives)</i> | 24 |
| 34 | <i>Andrei Andreyevich Markov, 1856-1922, Russe (St. Pétersbourg)</i> | 24 |
| 35 | <i>Binômes d'étudiants</i> | 25 |
| 36 | <i>Graphe des compétences des ouvriers</i> | 25 |
| 37 | <i>Carte d'Europe</i> | 26 |
| 38 | <i>Arthur Cayley, 1821-1895, anglais</i> | 27 |
| 39 | <i>Radars. Dames</i> | 27 |
| 40 | <i>Recette des crêpes flambées</i> | 28 |
| 41 | <i>Préparation des crêpes</i> | 29 |
| 42 | <i>Tableau des tâches</i> | 31 |
| 43 | <i>Diagramme des tâches</i> | 32 |
| 44 | <i>Diagramme de Gantt</i> | 33 |

Table des matières

| | |
|---|-----------|
| 1 Exemples introductifs, concept de graphe. | 1 |
| 1.1 Exemples de 1-graphes orientés. | 1 |
| 1.1.1 Division dans $[1; 10] \cap \mathbb{N}$. | 1 |
| 1.1.2 Les trois villas. | 2 |
| 1.1.3 Un arbre algébrique. | 3 |
| 1.1.4 Un algorithme. | 3 |
| 1.1.5 Interdépendance. | 4 |
| 1.2 Exemples de multigraphes. | 4 |
| 1.2.1 Un 3-graphe. | 4 |
| 1.3 Définitions. | 5 |
| 1.3.1 Chemin, chaîne, cycle, circuit. | 5 |
| 1.3.2 Graphe connexe, graphe non-connexe. | 6 |
| 1.3.3 Graphe non orienté. | 6 |
| 2 Représentations des graphes. | 7 |
| 2.1 Schémas et tables | 7 |
| 2.1.1 Diagramme sagittal. | 7 |
| 2.1.2 Tableau. | 7 |
| 2.1.3 Diagramme cartésien. | 7 |
| 2.2 Matrices associées ou matrices d'adjacence. | 7 |
| 2.2.1 Définition et exemple de matrice d'adjacence. | 7 |
| 2.2.2 Matrices booléennes. | 8 |
| 3 Problèmes de recherche de chemins. | 10 |
| 3.1 Problème du chemin entre deux points. | 10 |
| 3.1.1 Exemples classiques. | 10 |
| 3.2 Hamilton ou Euler ? | 10 |
| 3.2.1 Chemins hamiltoniens. | 10 |
| 3.2.2 Chaînes eulériennes. | 12 |
| 3.3 Chemins de longueur donnée. | 16 |
| 3.3.1 Longueur d'un chemin. | 16 |
| 3.3.2 Existence d'un chemin de longueur donnée. | 16 |
| 3.3.3 Nombres de chemins. | 17 |
| 3.3.4 Fermeture transitive d'un graphe. | 17 |
| 3.4 Valeur d'un chemin. | 18 |
| 3.4.1 Carte routière. | 18 |
| 3.4.2 Chemin de valeur minimale. | 18 |
| 3.4.3 Chemin de valeur maximale. | 19 |
| 3.4.4 Problèmes classiques et difficiles. | 19 |
| 4 Niveaux. | 20 |
| 5 Arbres et arborescences. | 20 |
| 5.1 Arbres. | 20 |
| 5.2 Arborescences. | 21 |
| 5.2.1 Arborescences binaires. | 21 |
| 6 Réseaux. | 22 |
| 6.1 Réseau de transport. | 22 |

| | | |
|----------|--|-----------|
| 7 | Autres utilisations. | 23 |
| 7.1 | Probabilités et chaînes de Markov. | 23 |
| 7.1.1 | 4 Faces consécutives au moins en n coups. | 23 |
| 7.2 | Problèmes d'affectation. | 24 |
| 7.2.1 | Couplage. | 24 |
| 7.3 | Coloration des sommets d'un graphe. | 26 |
| 7.3.1 | Problème des 4 couleurs. | 26 |
| 7.4 | Noyaux. | 27 |
| 7.4.1 | Nombre d'absorption. | 27 |
| 7.4.2 | Noyaux. | 28 |
| 7.5 | Ordonnancement de tâches. | 28 |
| 7.5.1 | Recette de crêpes. | 28 |
| 7.5.2 | Ordonnancement de tâches, activités sur les sommets. | 30 |
| A | Les graphes et le programme de BTS Informatique de gestion. | 34 |
| A.1 | Le programme. | 34 |
| A.2 | Domaines d'utilisation. | 34 |
| A.3 | Remarques | 34 |



Creative Commons : ATTRIBUTION - NONCOMMERCIAL - SHARE ALIKE
<http://creativecommons.org/licenses/by-nc-sa/3.0/>

GraphesBTS

Jean-Paul Davalan

<http://jeux-et-mathematiques.davalan.org/graphs/cours/index.html>