

# Graphe ch4 Problèmes et graphe

Pierre-Yves BISCHOFF

IUT Informatique Graphique

2020

# Sommaire

- 
- 1 Plus court chemin
  - 2 Coloration de graphe

# Sommaire de Plus court chemin

## 1 Plus court chemin

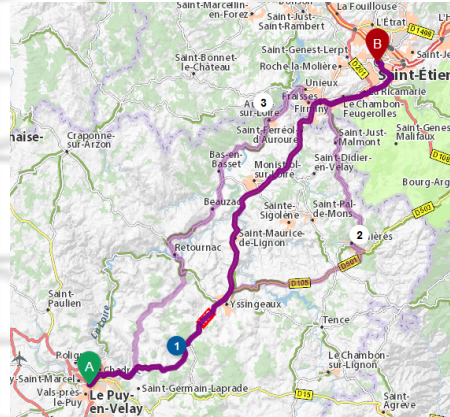
# Plus court chemin dans un graphe pondéré

## Exemple le GPS

Comment le GPS calcule-t-il la route la plus courte ?

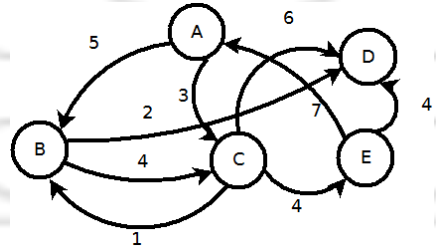
## Solution

Les données cartographiques contiennent la distance entre toutes les villes



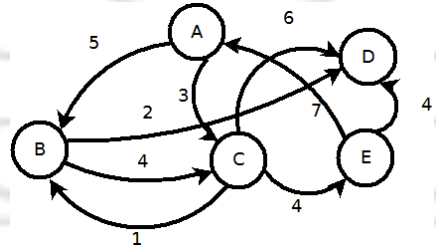
# Algorithme de Dijkstra initialisation

- ▶ Ajout d'un tableau de distance, visite, prédécesseur
- ▶ Initialisation des distance à l'infini, passage à 0, prédécesseur à -1
- ▶  $\text{distance}(\text{depart})=0$  et  $\text{visite}(\text{depart})=1$



## Résultat de l'initialisation

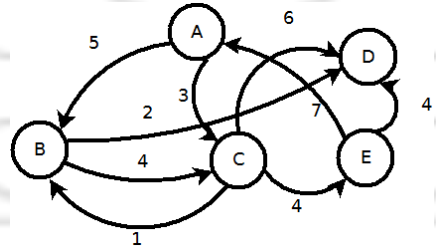
A	B	C	D	E
0 -1	$\infty$ -1	$\infty$ -1	$\infty$ -1	$\infty$ -1



# Algorithme de Dijkstra

tant que tous les sommets n'ont pas été visité

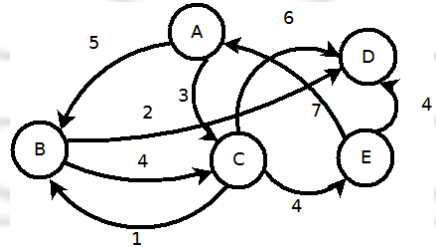
- ▶ Pour tous les sommets  $t$  voisins de  $s$
- ▶ si  $d(s) + \text{poids}(st) < d(t)$  Alors
  - $\text{pred}(t) = s$
  - $d(t) = d(s) + \text{poids}(st)$
  - $\text{visite}(t) = 0$
- ▶  $\text{visite}(s) = 1$



# Déroulé de l'algorithme de Dijkstra

On est au sommet A  
ses voisins sont B et C

A	B	C	D	E
0 -1	$\infty$ -1	$\infty$ -1	$\infty$ -1	$\infty$ -1
0 -1	5 A	3 A	$\infty$ -1	$\infty$ -1

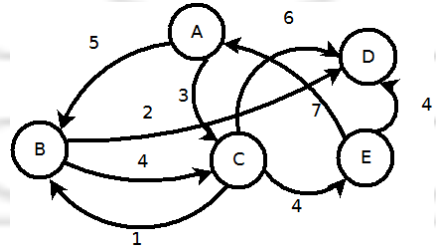




# Algorithme de Dijkstra

tant que tous les sommets n'ont pas été visité

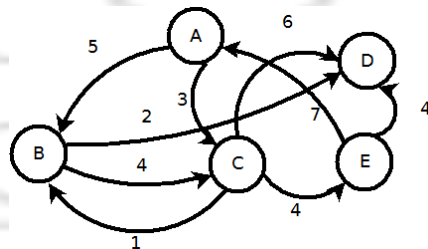
- ▶ Pour tous les sommets  $t$  voisins de  $s$
- ▶ si  $d(s) + \text{poids}(st) < d(t)$  Alors
  - $\text{pred}(t) = s$
  - $d(t) = d(s) + \text{poids}(st)$
  - $\text{visite}(t) = 0$
- ▶  $\text{passage}(s) = 1$



# Déroulé de l'algorithme de Dijkstra

On est au sommet B  
ses voisins sont C et D

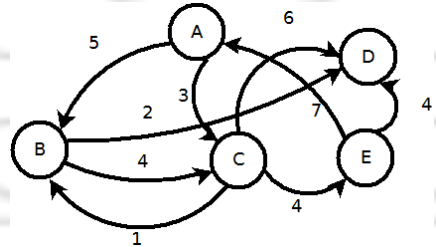
A	B	C	D	E
0 -1	$\infty$ -1	$\infty$ -1	$\infty$ -1	$\infty$ -1
0 -1	5 A	3 A	$\infty$ -1	$\infty$ -1
0 -1	5 A	3 A	7 B	$\infty$ -1



# Algorithme de Dijkstra

tant que tous les sommets n'ont pas été visité

- ▶ Pour tous les sommets  $t$  voisins de  $s$
- ▶ si  $d(s) + \text{poids}(st) < d(t)$  Alors
  - $\text{pred}(t) = s$
  - $d(t) = d(s) + \text{poids}(st)$
  - $\text{visite}(t) = 0$
- ▶  $\text{passage}(s) = 1$

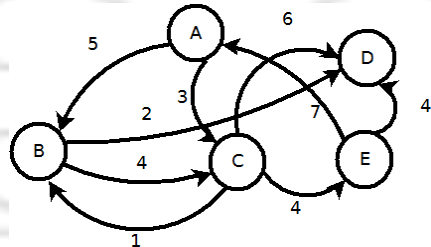


# Déroulé de l'algorithme de Dijkstra

On est au sommet C  
ses voisins sont B, D et E

A	B	C	D	E
0 -1	$\infty$ -1	$\infty$ -1	$\infty$ -1	$\infty$ -1
0 -1	5 A	3 A	$\infty$ -1	$\infty$ -1
0 -1	5 A	3 A	7 B	$\infty$ -1
0 -1	4 C	3 A	7 B	7 C

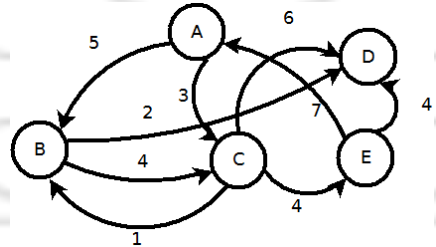
Comme B a été modifié on le remet non visité



# Algorithme de Dijkstra

tant que tous les sommets n'ont pas été visité

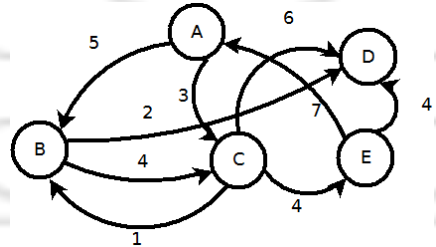
- ▶ Pour tous les sommets  $t$  voisins de  $s$
- ▶ si  $d(s) + \text{poids}(st) < d(t)$  Alors
  - $\text{pred}(t) = s$
  - $d(t) = d(s) + \text{poids}(st)$
  - $\text{visite}(t) = 0$
- ▶  $\text{passage}(s) = 1$



# Déroulé de l'algorithme de Dijkstra

On va reparcourir B

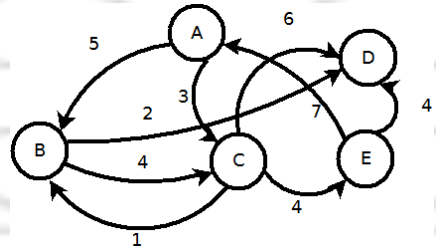
A	B	C	D	E
0 -1	$\infty$ -1	$\infty$ -1	$\infty$ -1	$\infty$ -1
0 -1	5 A	3 A	$\infty$ -1	$\infty$ -1
0 -1	5 A	3 A	7 B	$\infty$ -1
0 -1	4 C	3 A	7 B	7 C
0 -1	4 C	3 A	6 B	7 C



# Déroulé de l'algorithme de Dijkstra

On va parcourir D puis E  
sans aucun changement

A	B	C	D	E
0 -1	$\infty$ -1	$\infty$ -1	$\infty$ -1	$\infty$ -1
0 -1	5 A	3 A	$\infty$ -1	$\infty$ -1
0 -1	5 A	3 A	7 B	$\infty$ -1
0 -1	4 C	3 A	7 B	7 C
0 -1	4 C	3 A	6 B	7 C



# Remarques

## Algorithme de Dijkstra : 1959

On calcule la distance minimale entre le départ et tous les sommets du graphe  
l'algorithme originel était en  $O(n^2)$

## Modification par Tarjan : 1987

$O(n \log(n))$

## Contraction hiérarchique

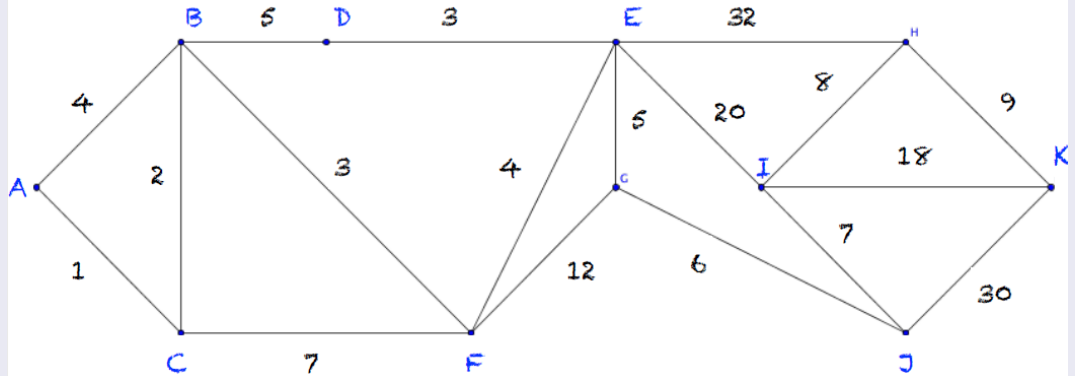
Mémoire de master par Geisberger 2008

- ▶ Simplicity is prerequisite for reliability
- ▶ The question of whether Machines Can Think... is about as relevant as the question of whether Submarines Can Swim
- ▶ Object-oriented programming is an exceptionally bad idea which could only have originated in California
- ▶ Simplicity is a great virtue but it requires hard work to achieve it and education to appreciate it.



# Test

À vous de jouer départ de A



# Déroulé de l'algorithme de Dijkstra

Sommet	A	B	C	D	E	F	G	H	I	J	K
A	0 -1	4 A	1 A	$\infty$ -1	$\infty$ -1	$\infty$ -1	$\infty$ -1	$\infty$ -1	$\infty$ -1	$\infty$ -1	$\infty$ -1
C	0 -1	3 C	1 A	$\infty$ -1	$\infty$ -1	8 C	$\infty$ -1	$\infty$ -1	$\infty$ -1	$\infty$ -1	$\infty$ -1
B	0 -1	3 C	1 A	8 B	$\infty$ -1	6 B	$\infty$ -1	$\infty$ -1	$\infty$ -1	$\infty$ -1	$\infty$ -1
F	0 -1	3 C	1 A	8 B	10 F	6 B	22 F	$\infty$ -1	$\infty$ -1	$\infty$ -1	$\infty$ -1
E	0 -1	3 C	1 A	8 B	10 F	6 B	15 E	42 E	30 E	$\infty$ -1	$\infty$ -1
G	0 -1	3 C	1 A	8 B	10 F	6 B	15 E	42 E	30 E	21 G	$\infty$ -1
J	0 -1	3 C	1 A	8 B	10 F	6 B	15 E	42 E	28 J	21 G	51 J
I	0 -1	3 C	1 A	8 B	10 F	6 B	15 E	36 I	28 J	21 G	46 I
H	0 -1	3 C	1 A	8 B	10 F	6 B	15 E	36 I	28 J	21 G	45 H

# Sommaire de Coloration de graphe

## 2 Coloration de graphe

# Problème de géographie

# Combien de couleurs faut-il pour colorer une carte ?

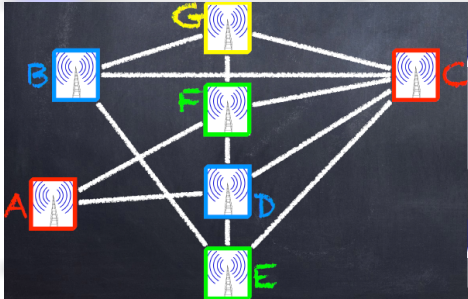
On cherche le nombre minimum de couleurs pour colorer une carte telle que deux régions adjacentes ne soient pas de la même couleur

# Nombre chromatique d'un graphe

le nombre minimal de couleurs qu'il faut employer pour colorer chacun des sommets de ce graphe de sorte que deux sommets adjacents quelconques ne soient jamais de la même couleur



## minorant du nombre Chromatique



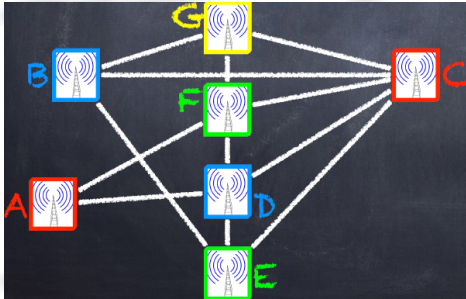
### Minorant

Le nombre chromatique  $\chi$  d'un graphe  $G$  est supérieur ou égal à l'ordre  $n$  du sous-graphe complet le plus grand de  $G$

### sous graphe complet

Ici le sous graphe complet le plus grand est BCE

# majorant du nombre Chromatique



## Majorant

Le nombre chromatique  $\chi$  d'un graphe  $G$  est inférieur ou égal à  $d + 1$  le plus grand degré d'un sommet de  $G$

## sous graphe complet

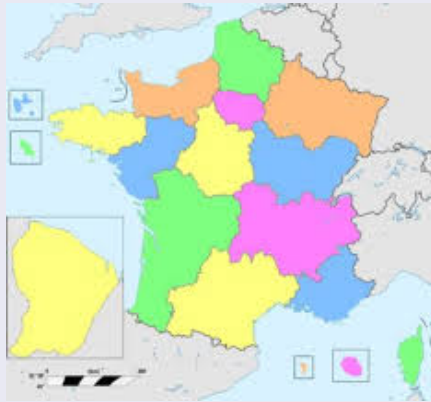
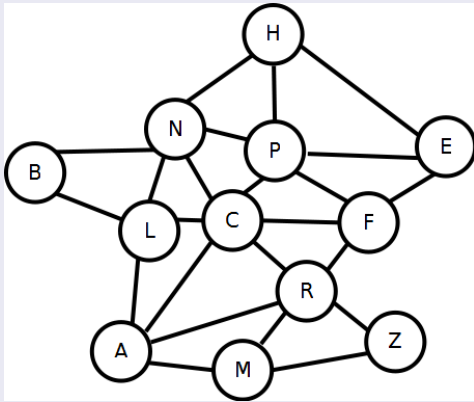
Ici le sommet  $C$  est de plus grand degré 5 donc  
 $3 \leq \chi \leq 6$

# Algorithme de Welsh et Powell

- ① Ranger les sommets par ordre de degrés décroissants
- ② Attribuer au premier sommet (A) de la liste une couleur
- ③ Suivre la liste en attribuant la même couleur au premier sommet (B) qui ne soit pas adjacent à (A)
- ④ Suivre (si possible) la liste jusqu'au prochain sommet (C) qui ne soit adjacent ni à A ni à B ...
- ⑤ Prendre une nouvelle couleur pour le premier sommet non coloré
- ⑥ Répéter 3 à 4 puis refaire 5

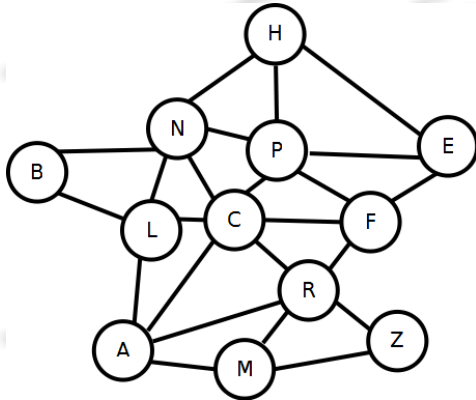
# Test

À vous de jouer pour colorer la France





# initialisation de Welsh et Powell



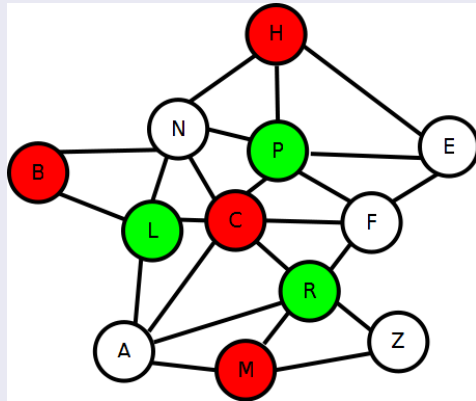
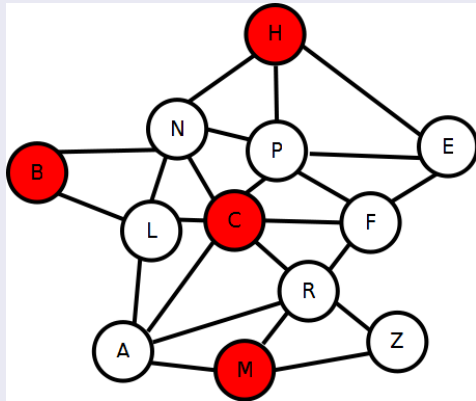
Sommet	degré
C	6
P	5
N	5
R	5
L	4
F	4
A	4
H	3
E	3
M	3
B	2
Z	2

# Algorithme de Welsh et Powell

- ① Ranger les sommets par ordre de degrés décroissants
- ② Attribuer au premier sommet (A) de la liste une couleur
- ③ Suivre la liste en attribuant la même couleur au premier sommet (B) qui ne soit pas adjacent à (A)
- ④ Suivre (si possible) la liste jusqu'au prochain sommet (C) qui ne soit adjacent ni à A ni à B ...
- ⑤ Prendre une nouvelle couleur pour le premier sommet non coloré
- ⑥ Répéter 3 à 4 puis refaire 5

# Test

À vous de jouer pour colorer la France



# Test

À vous de jouer pour colorer la France

