Parcours de graphes

Florent Foucaud - Malika More - Thibault Ralet
Carine Simon - Thierry Trévisan

1A - BUT Info - UCA

R207 Graphes

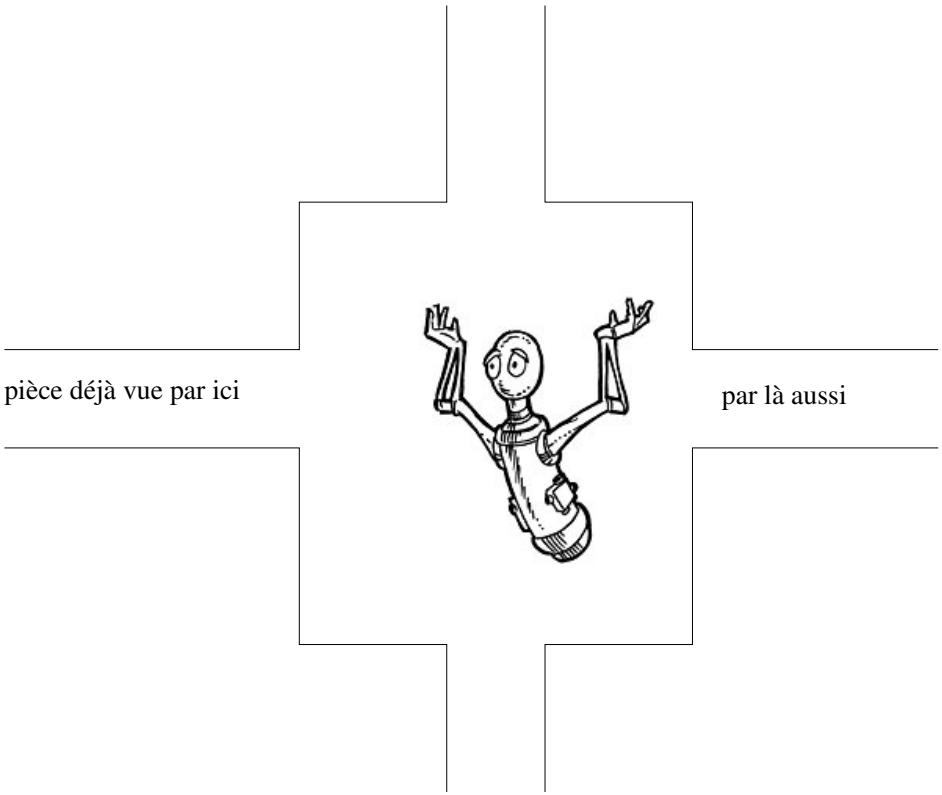
Année 2021-2022

Sommaire

1. Introduction
2. Exemple
3. File et pile
4. Parcours en largeur
   - Rappel sur les files
   - Algorithme
   - Arbre de parcours et calcul des distances
5. Parcours en profondeur
   - Rappel sur les piles
   - Algorithme
   - Arbre de parcours
6. Pour finir

Le robot explorateur

Un robot a pour mission d'explorer un bâtiment d'importance archéologique. Ce bâtiment a été englouti par une coulée de boue. Au cours des âges, la boue a séché et forme des blocs compacts difficiles à percer.

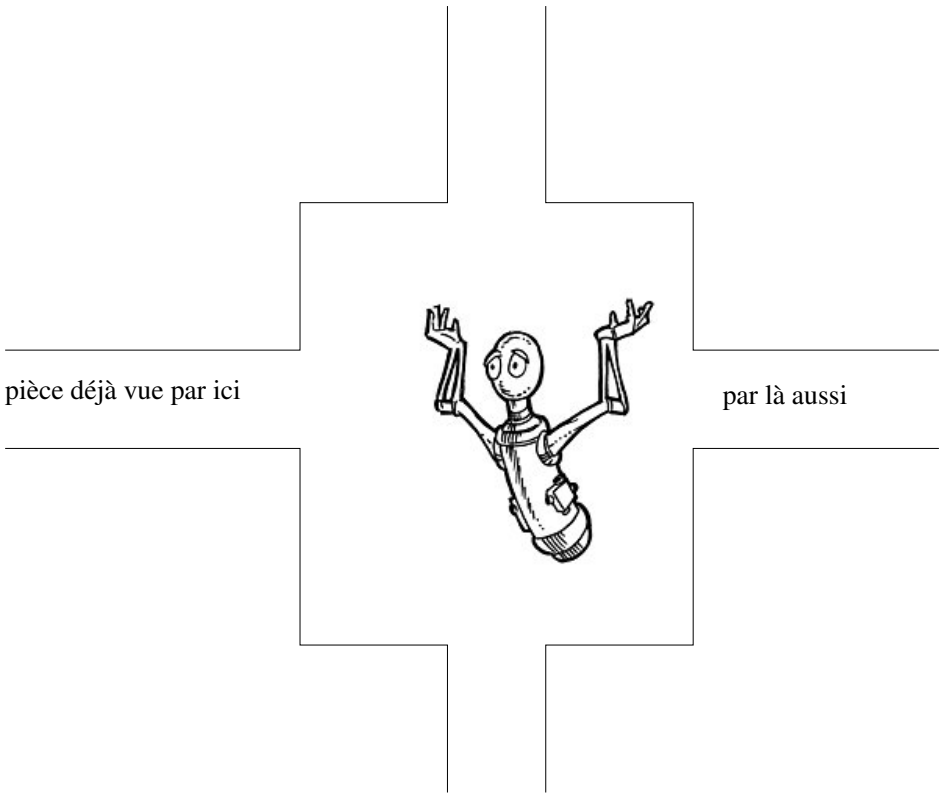


Les balises

Le robot dispose de petites balises pour marquer chaque pièce visitée. Ainsi depuis la pièce où il se trouve, il peut détecter quels couloirs mènent à une pièce balisée.

Question

Quelle stratégie pourrait-il adopter afin d'explorer entièrement le bâtiment en minimisant le nombre de couloirs traversés (puisqu'il faut déblayer la boue avant) ?

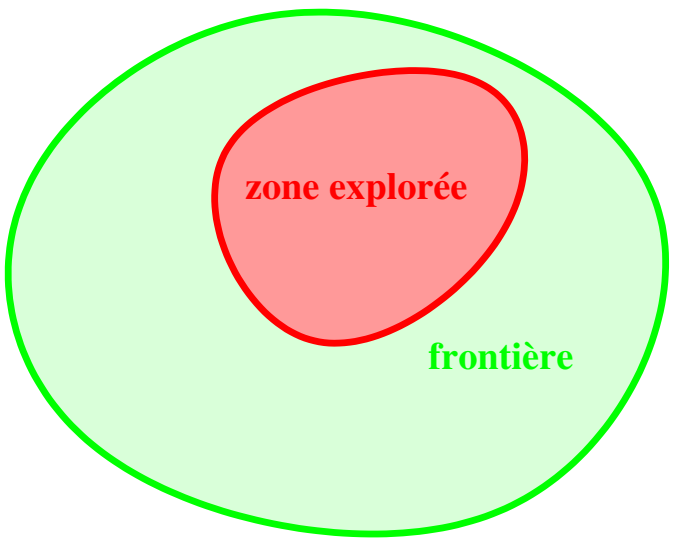


Parcours d'un graphe

Stratégie

On garde en mémoire l'ensemble des sommets connus, partitionné en deux :

- d'une part, ceux qu'on a complètement visités, qui forment la zone explorée ; et,
- d'autre part, ceux qu'on n'a pas encore entièrement visités, qui forment la frontière de la zone explorée.

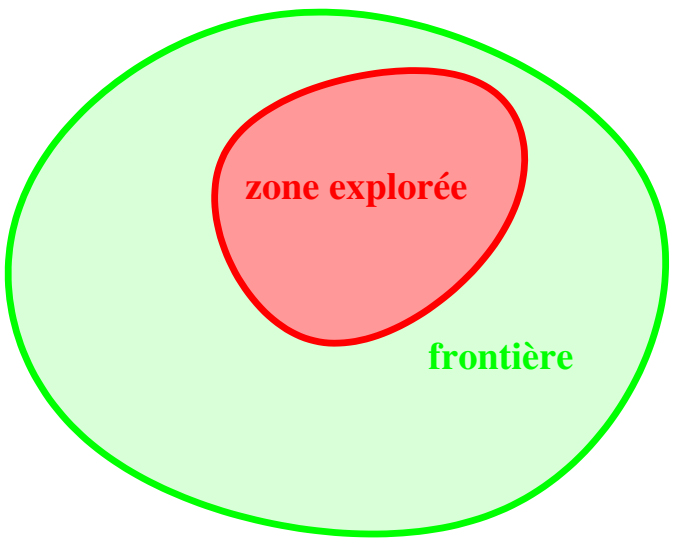


Parcours d'un graphe

Méthode

On se place sur un sommet v de la frontière. Deux cas sont possibles :

- ou bien tous les voisins de v sont déjà connus (zone explorée + frontière).
  - Dans ce cas, on ajoute v à la zone explorée.
- ou bien v a encore au moins un voisin u dans l'inconnu.
  - Dans ce cas, on *marque l'arête* $\{v, u\}$ puis on ajoute u à la frontière.

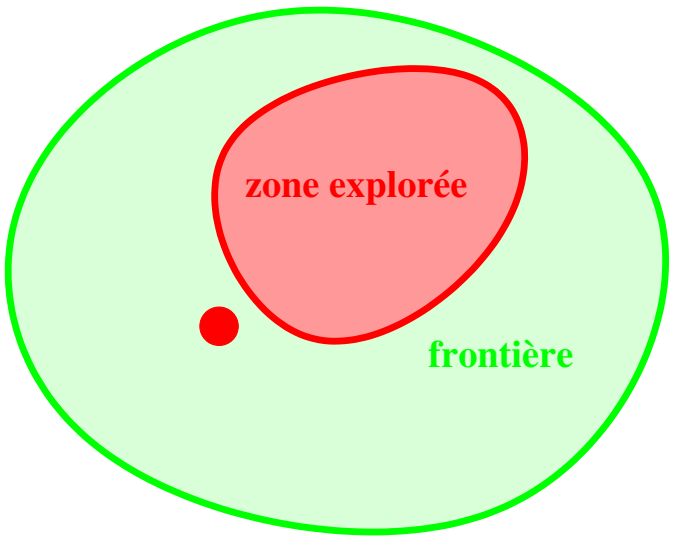


Parcours d'un graphe

Méthode

On se place sur un sommet v de la frontière. Deux cas sont possibles :

- ou bien tous les voisins de v sont déjà connus (zone explorée + frontière).
  - Dans ce cas, on ajoute v à la zone explorée.
- ou bien v a encore au moins un voisin u dans l'inconnu.
  - Dans ce cas, on *marque l'arête* $\{v, u\}$ puis on ajoute u à la frontière.

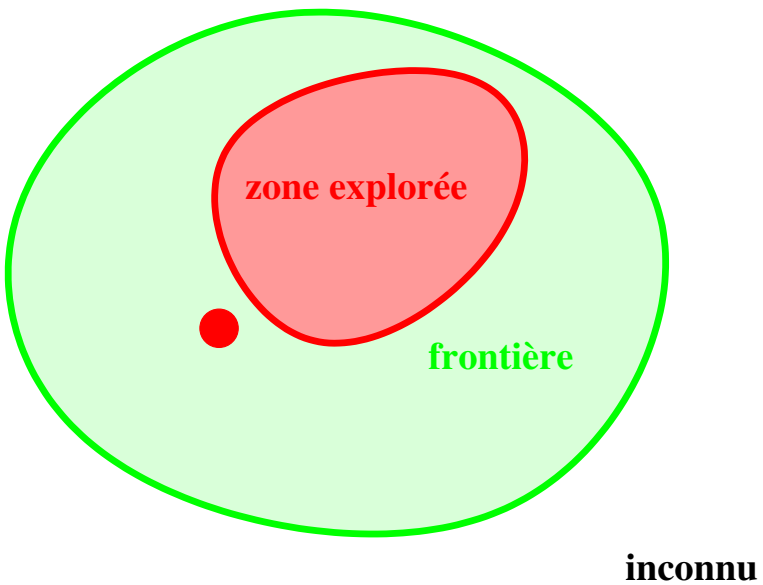


Parcours d'un graphe

Méthode

On se place sur un sommet v de la frontière. Deux cas sont possibles :

- ou bien tous les voisins de v sont déjà connus (zone explorée + frontière).
  - Dans ce cas, on ajoute v à la zone explorée.
- ou bien v a encore au moins un voisin u dans l'inconnu.
  - Dans ce cas, on *marque l'arête* $\{v, u\}$ puis on ajoute u à la frontière.

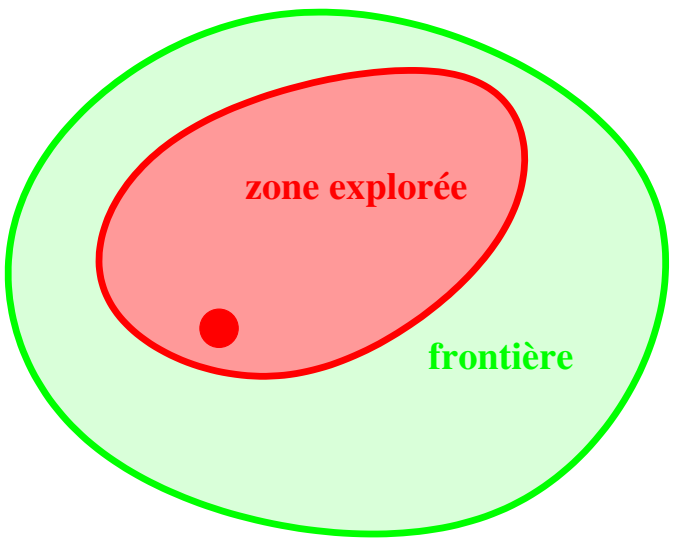


Parcours d'un graphe

Méthode

On se place sur un sommet v de la frontière. Deux cas sont possibles :

- ou bien tous les voisins de v sont déjà connus (zone explorée + frontière).
  - Dans ce cas, on ajoute v à la zone explorée.
- ou bien v a encore au moins un voisin u dans l'inconnu.
  - Dans ce cas, on *marque l'arête* $\{v, u\}$ puis on ajoute u à la frontière.

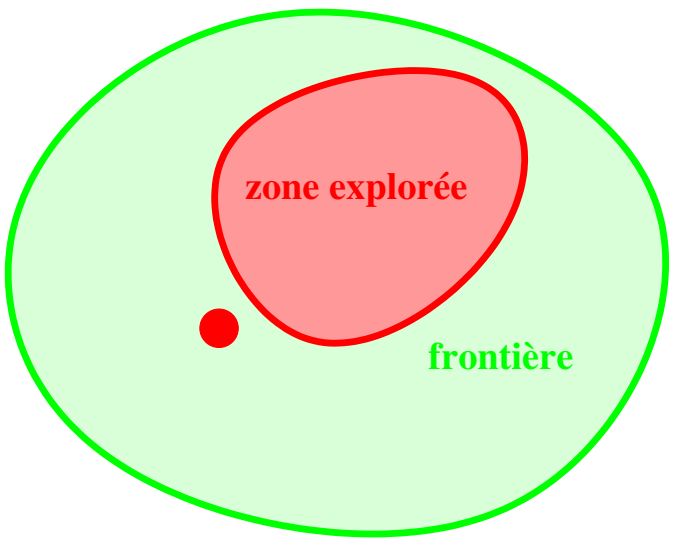


Parcours d'un graphe

Méthode

On se place sur un sommet v de la frontière. Deux cas sont possibles :

- ou bien tous les voisins de v sont déjà connus (zone explorée + frontière).
  - Dans ce cas, on ajoute v à la zone explorée.
- ou bien v a encore au moins un voisin u dans l'inconnu.
  - Dans ce cas, on *marque l'arête* $\{v, u\}$ puis on ajoute u à la frontière.

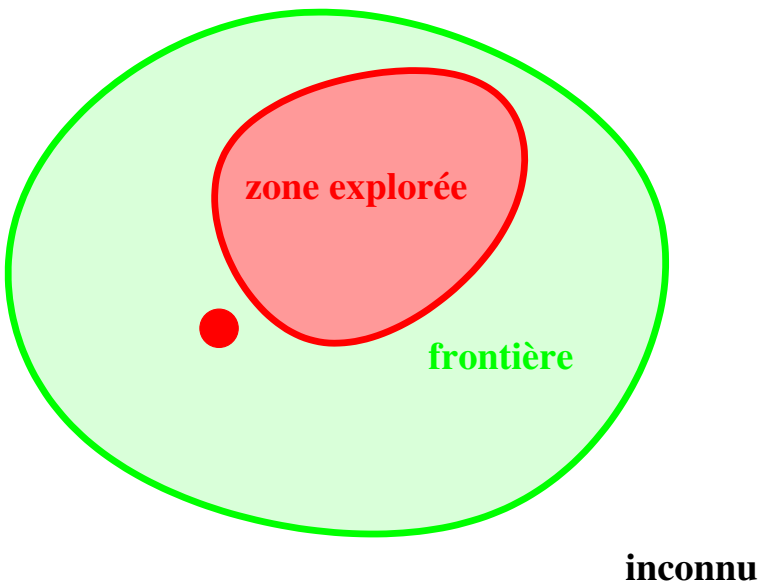


Parcours d'un graphe

Méthode

On se place sur un sommet v de la frontière. Deux cas sont possibles :

- ou bien tous les voisins de v sont déjà connus (zone explorée + frontière).
  - Dans ce cas, on ajoute v à la zone explorée.
- ou bien v a encore au moins un voisin u dans l'inconnu.
  - Dans ce cas, on *marque l'arête* $\{v, u\}$ puis on ajoute u à la frontière.

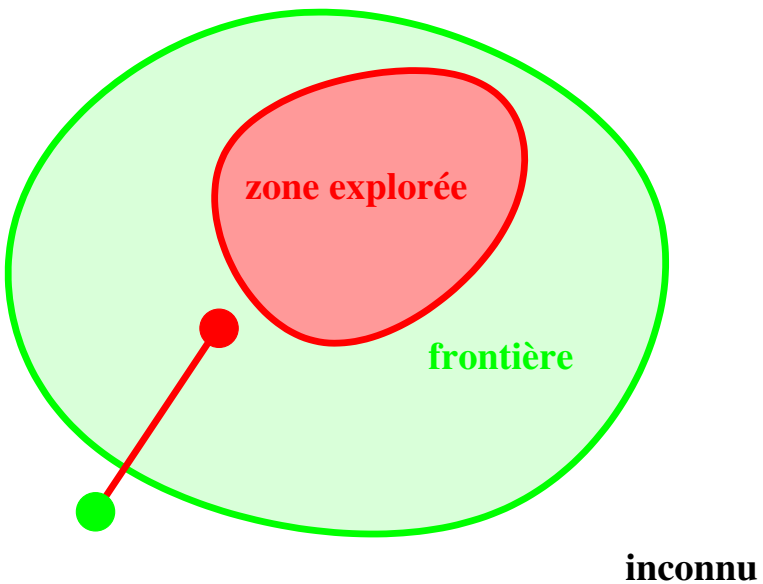


Parcours d'un graphe

Méthode

On se place sur un sommet v de la frontière. Deux cas sont possibles :

- ou bien tous les voisins de v sont déjà connus (zone explorée + frontière).
  - Dans ce cas, on ajoute v à la zone explorée.
- ou bien v a encore au moins un voisin u dans l'inconnu.
  - Dans ce cas, on *marque l'arête* $\{v, u\}$ puis on ajoute u à la frontière.

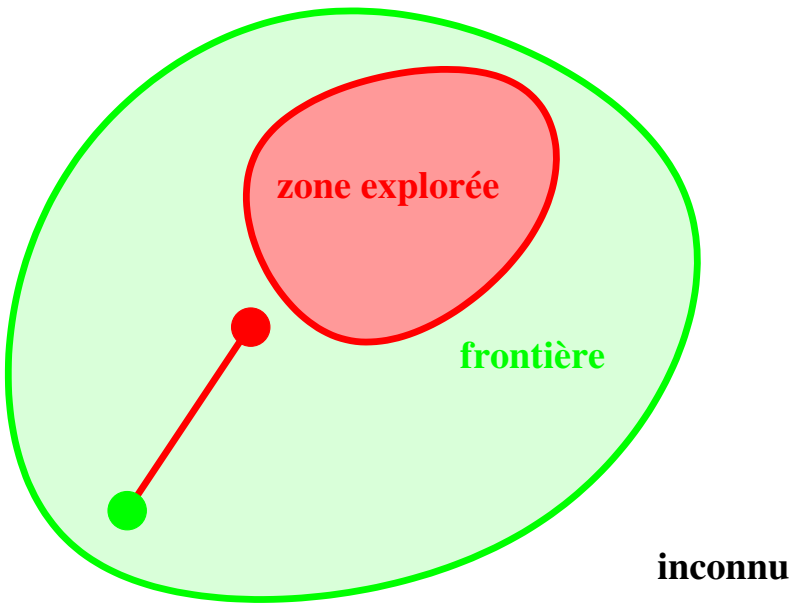


Parcours d'un graphe

Méthode

On se place sur un sommet v de la frontière. Deux cas sont possibles :

- ou bien tous les voisins de v sont déjà connus (zone explorée + frontière).
  - Dans ce cas, on ajoute v à la zone explorée.
- ou bien v a encore au moins un voisin u dans l'inconnu.
  - Dans ce cas, on *marque l'arête* $\{v, u\}$ puis on ajoute u à la frontière.

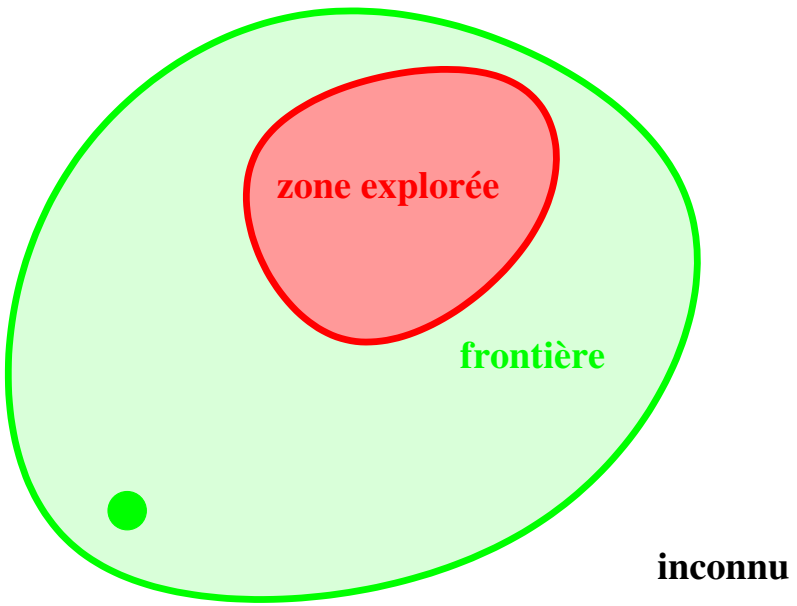


Parcours d'un graphe

Méthode

On se place sur un sommet v de la frontière. Deux cas sont possibles :

- ou bien tous les voisins de v sont déjà connus (zone explorée + frontière).
  - Dans ce cas, on ajoute v à la zone explorée.
- ou bien v a encore au moins un voisin u dans l'inconnu.
  - Dans ce cas, on *marque l'arête* $\{v, u\}$ puis on ajoute u à la frontière.

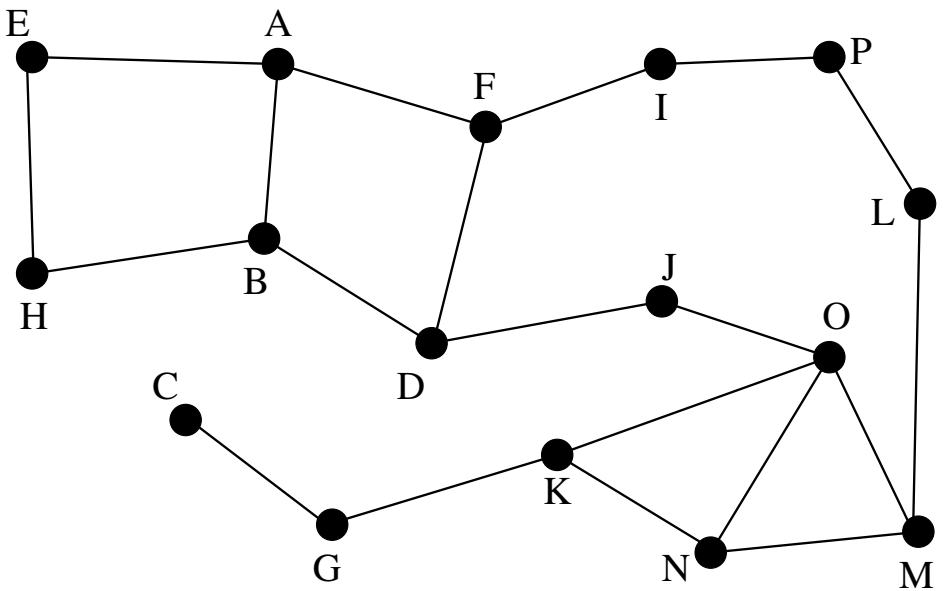


Sommaire

1. Introduction
2. Exemple
3. File et pile
4. Parcours en largeur
   - Rappel sur les files
   - Algorithme
   - Arbre de parcours et calcul des distances
5. Parcours en profondeur
   - Rappel sur les piles
   - Algorithme
   - Arbre de parcours
6. Pour finir

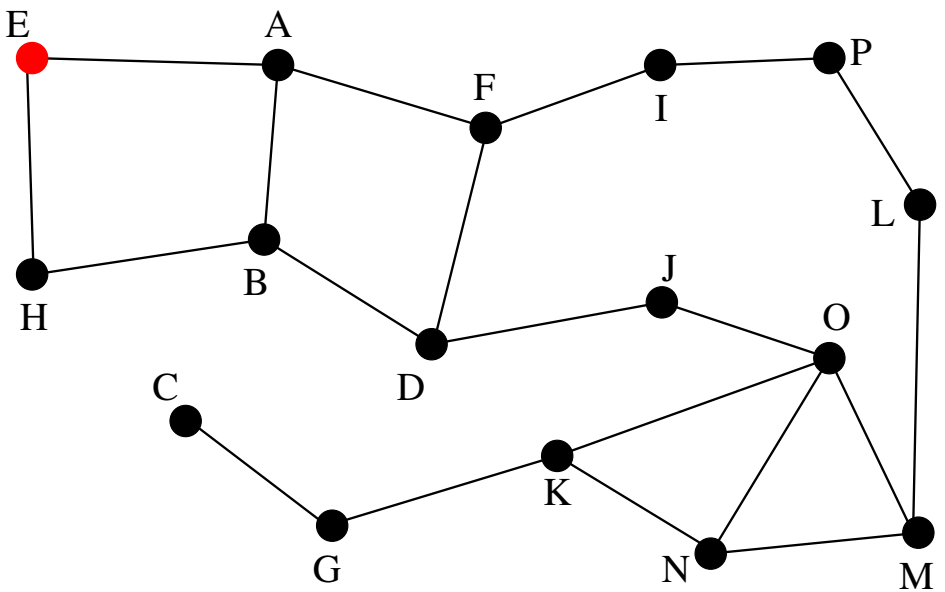
Parcours quelconque

▸ Fin Animation



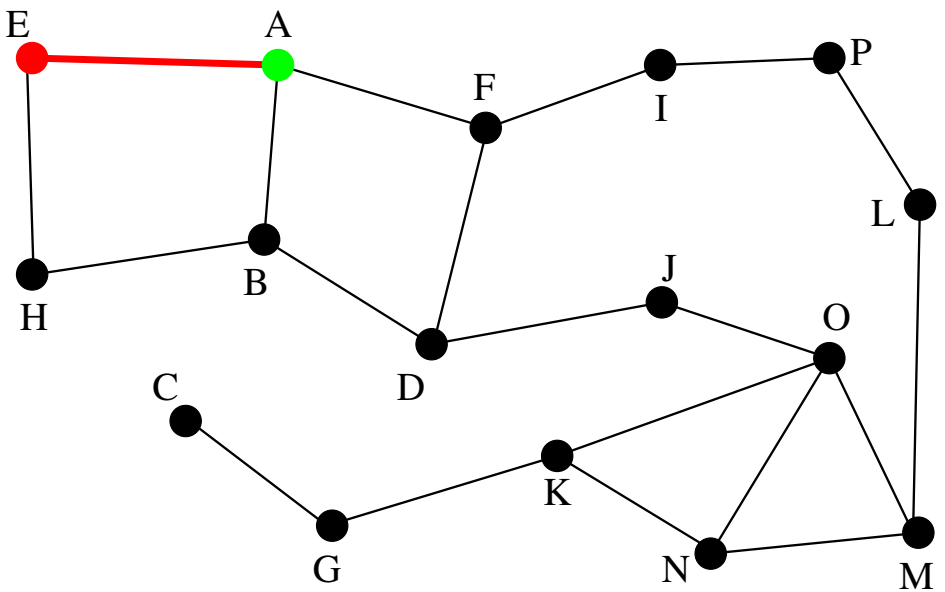
Parcours quelconque

▸ Fin Animation



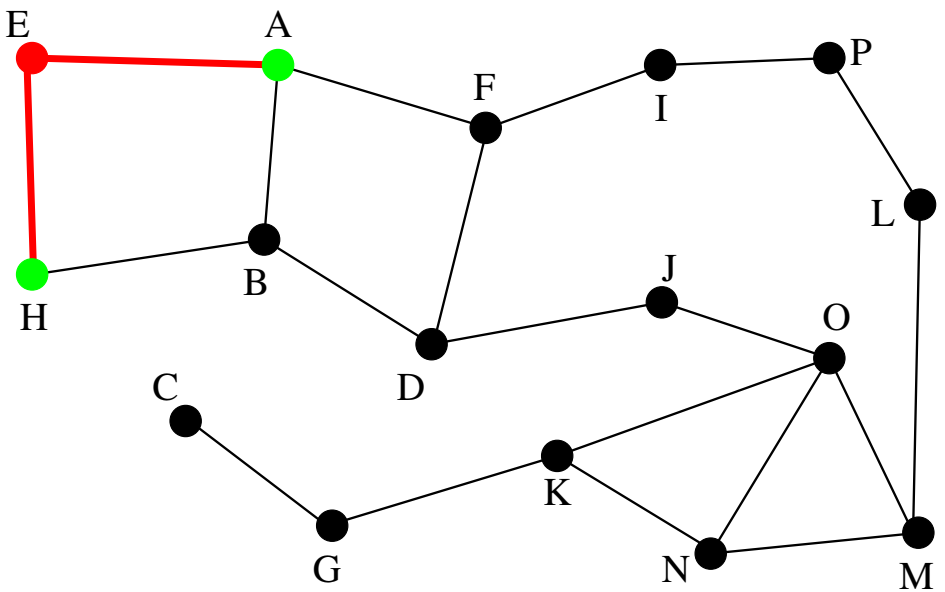
Parcours quelconque

▸ Fin Animation



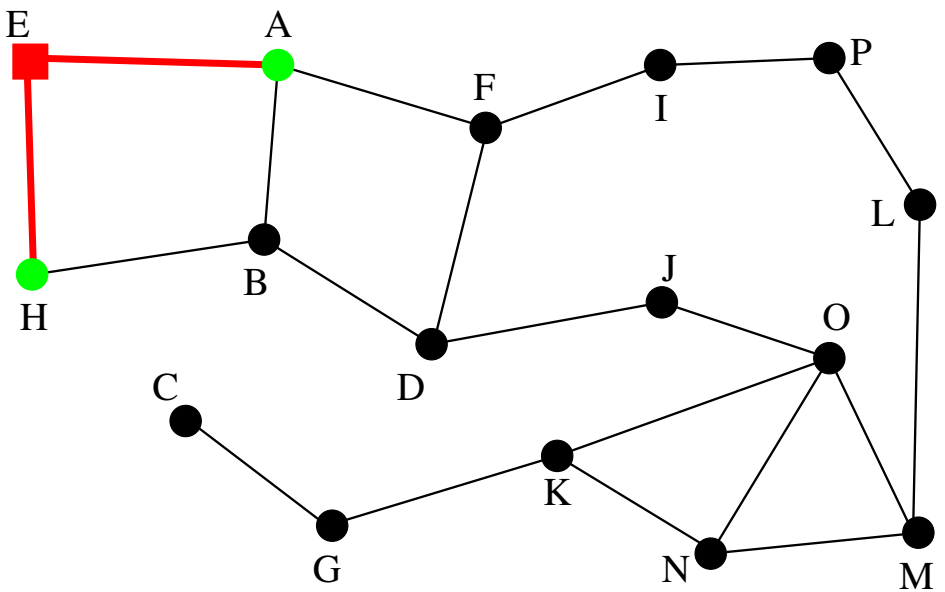
Parcours quelconque

▸ Fin Animation



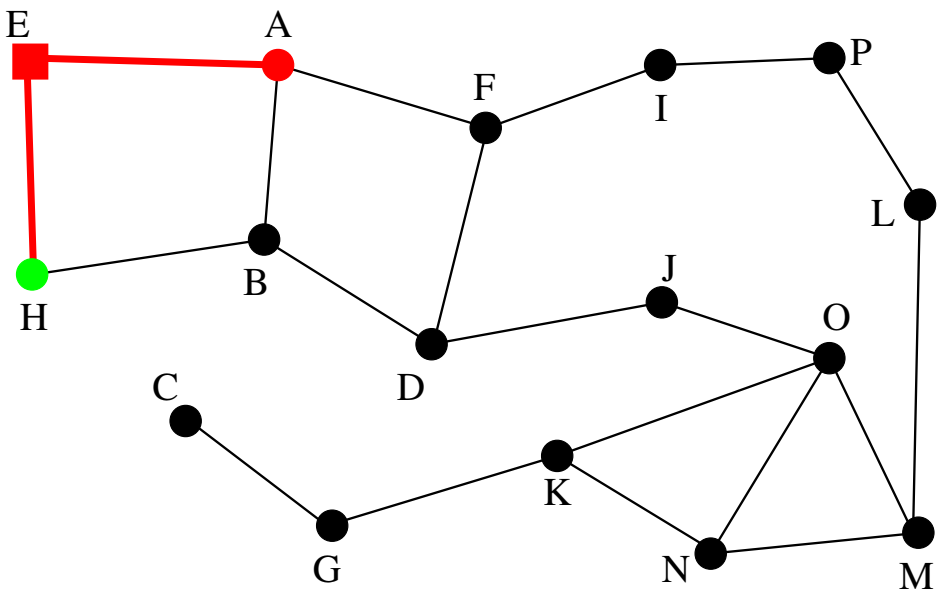
Parcours quelconque

▸ Fin Animation



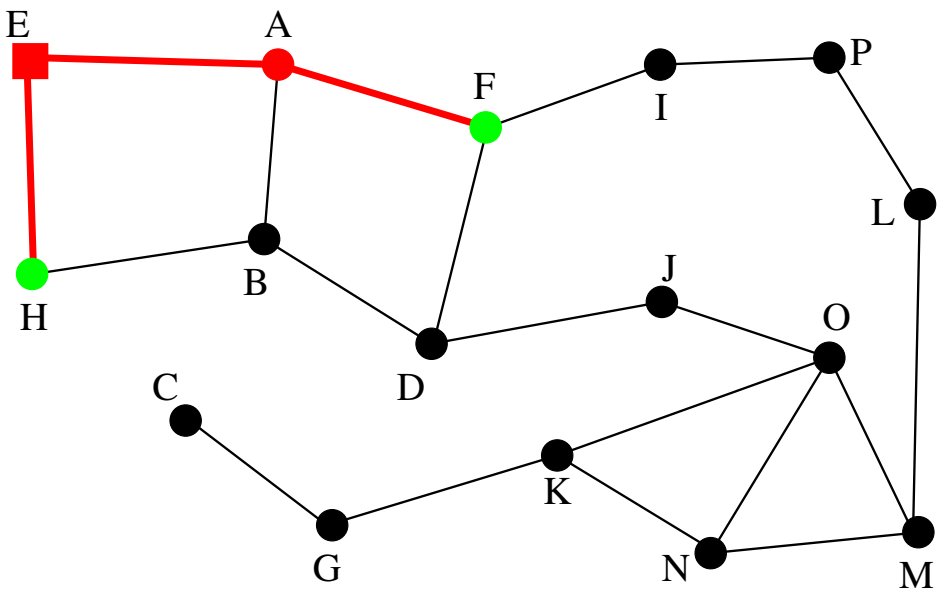
Parcours quelconque

▸ Fin Animation



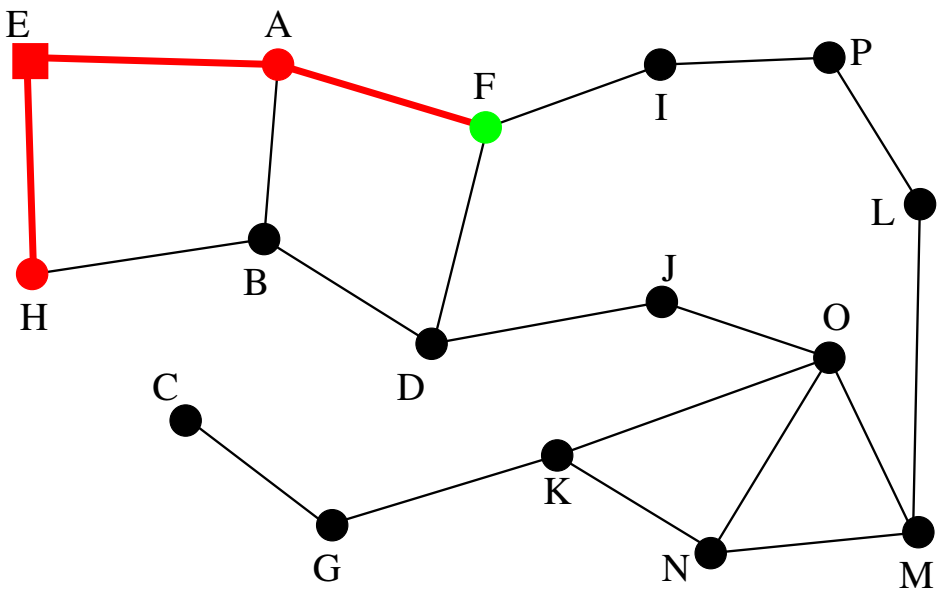
Parcours quelconque

▸ Fin Animation



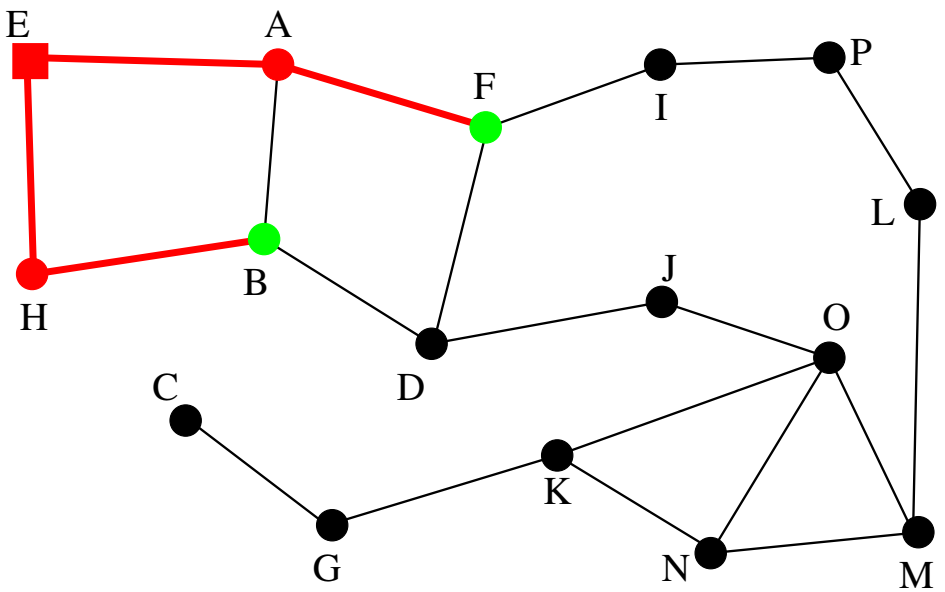
Parcours quelconque

▸ Fin Animation



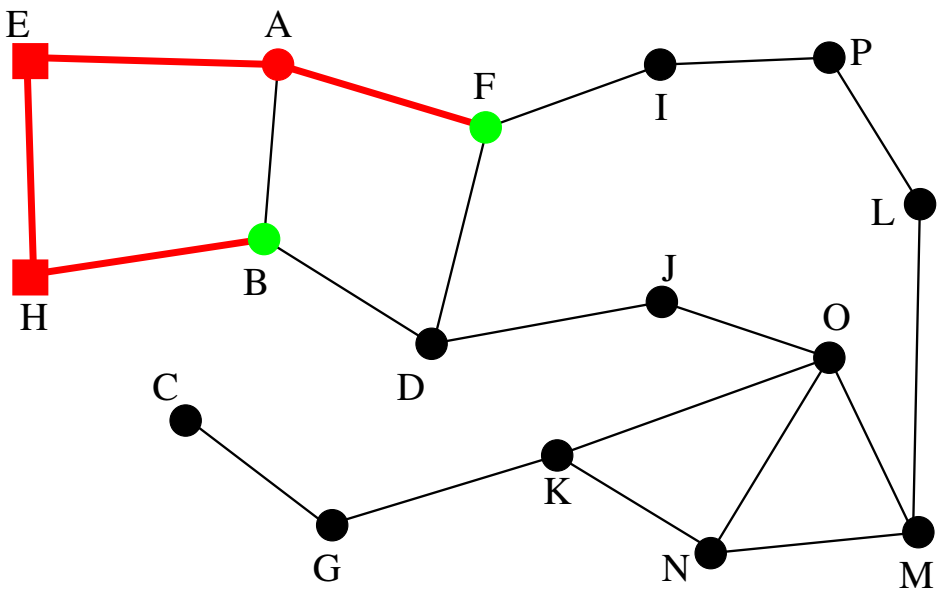
Parcours quelconque

▸ Fin Animation



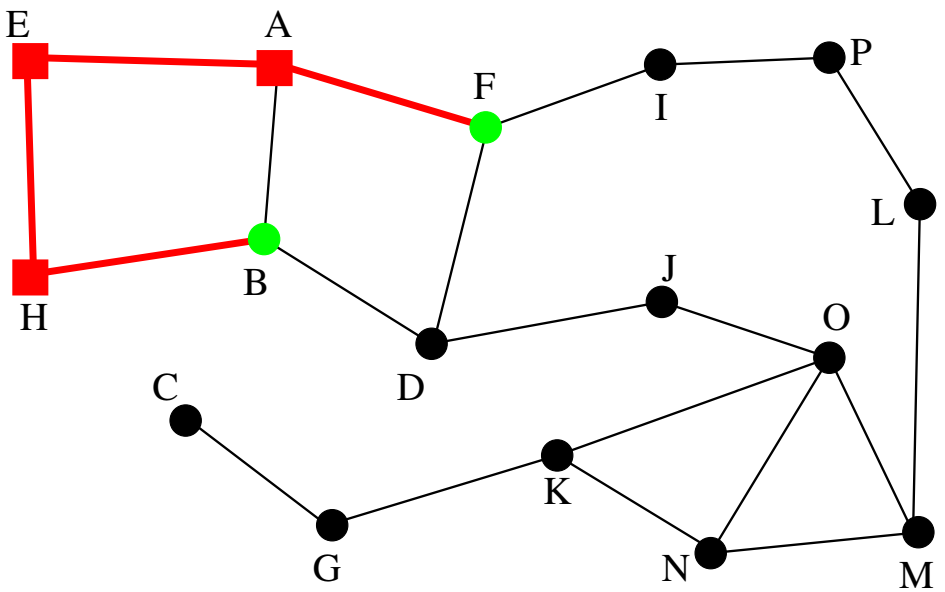
Parcours quelconque

▸ Fin Animation



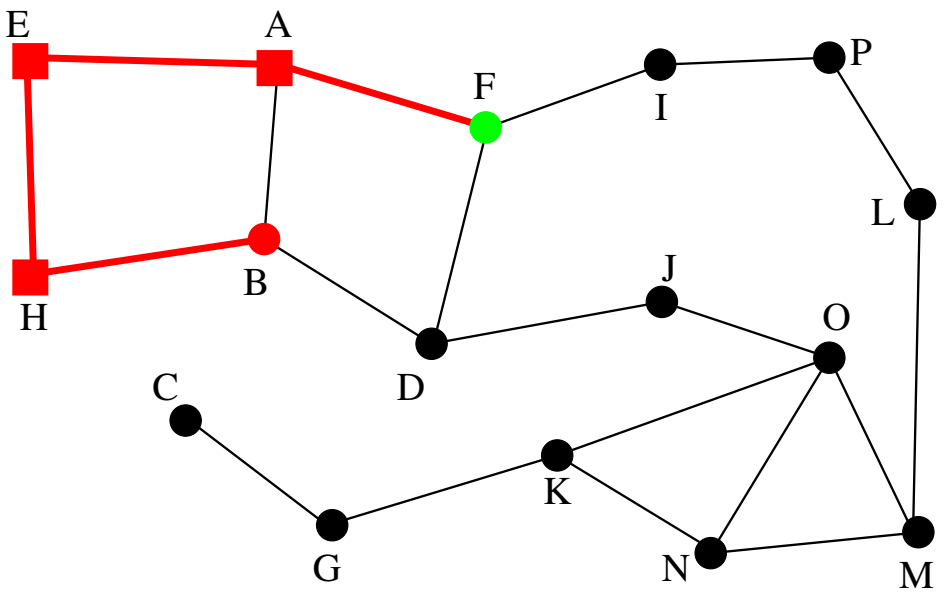
Parcours quelconque

▸ Fin Animation



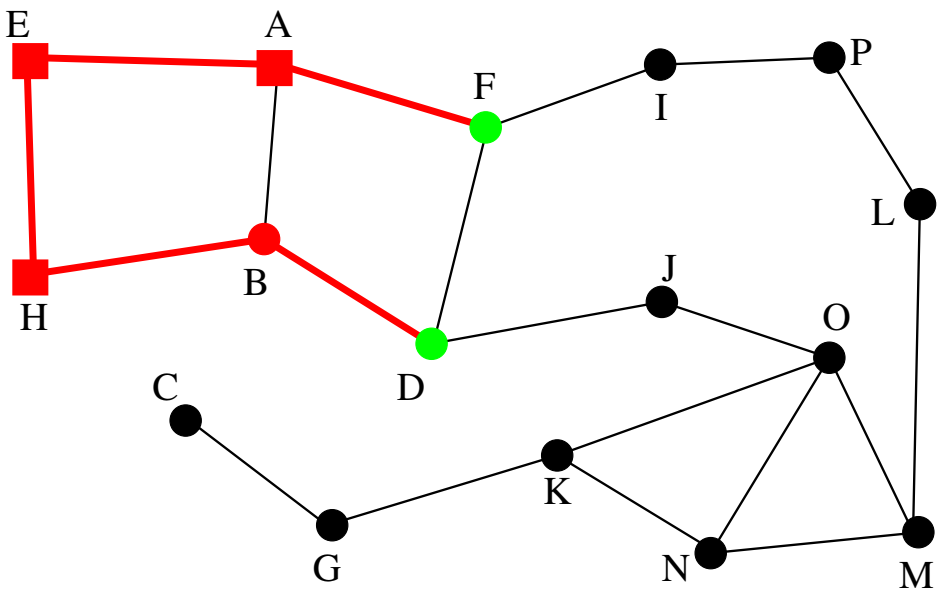
Parcours quelconque

▸ Fin Animation



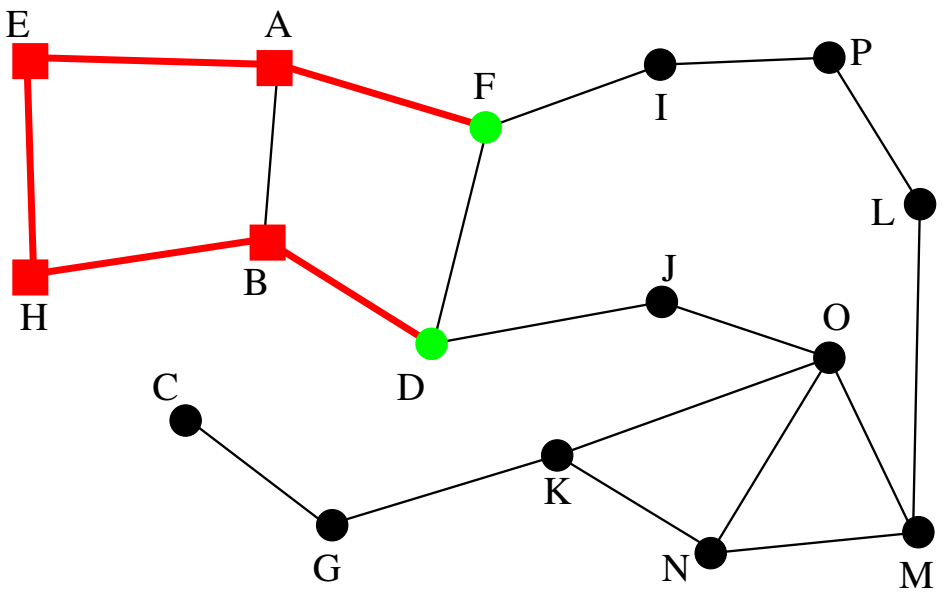
Parcours quelconque

▸ Fin Animation



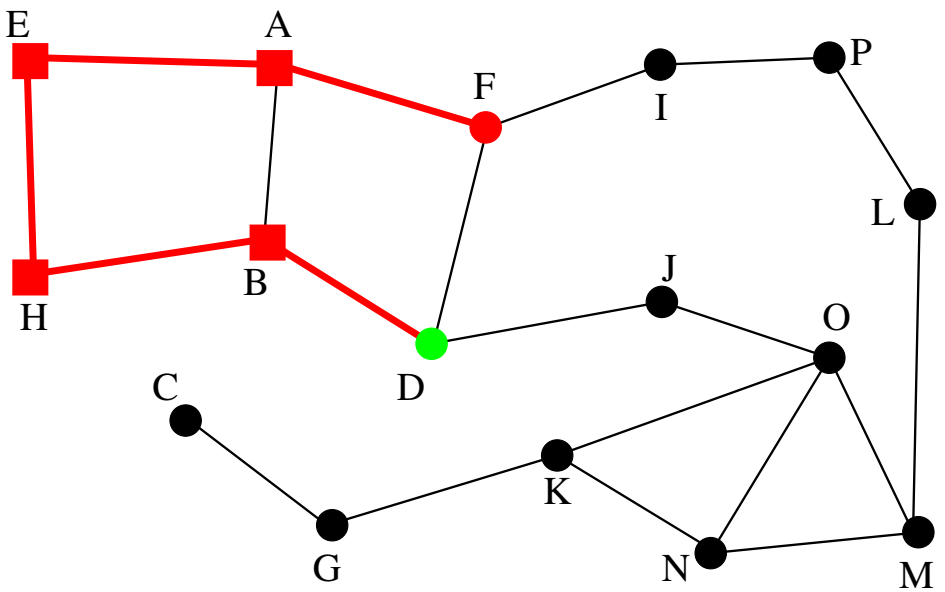
Parcours quelconque

▸ Fin Animation



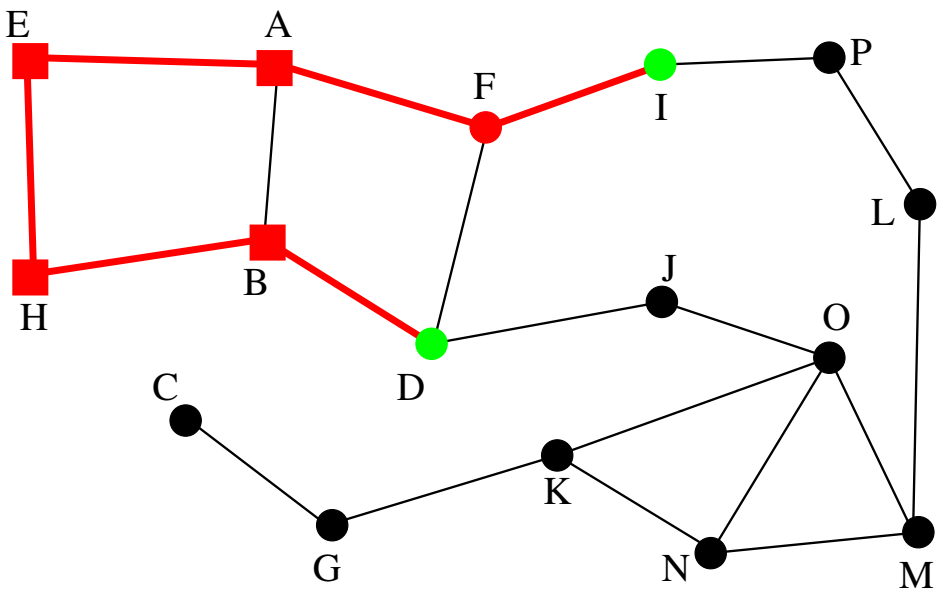
Parcours quelconque

▸ Fin Animation



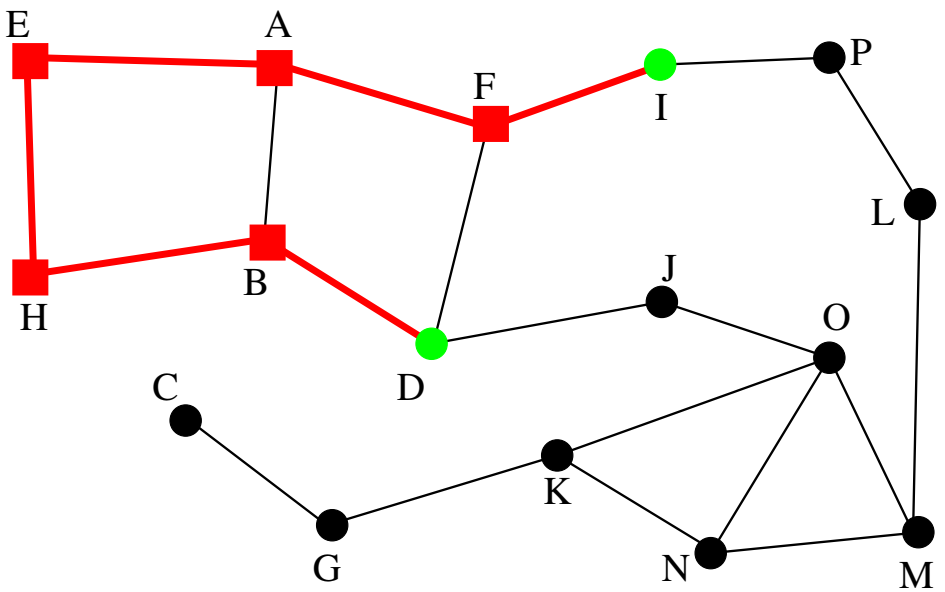
Parcours quelconque

▸ Fin Animation



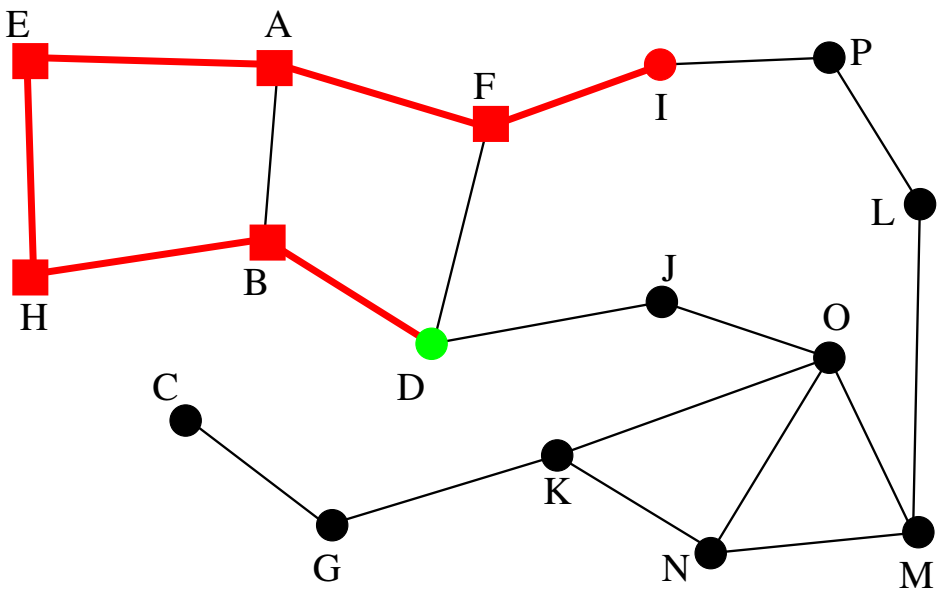
Parcours quelconque

▸ Fin Animation



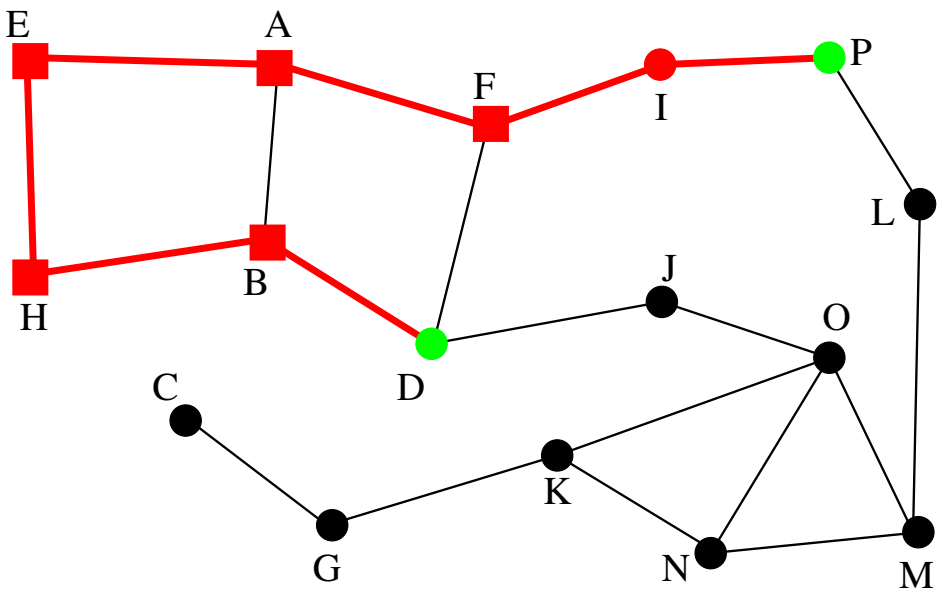
Parcours quelconque

▸ Fin Animation



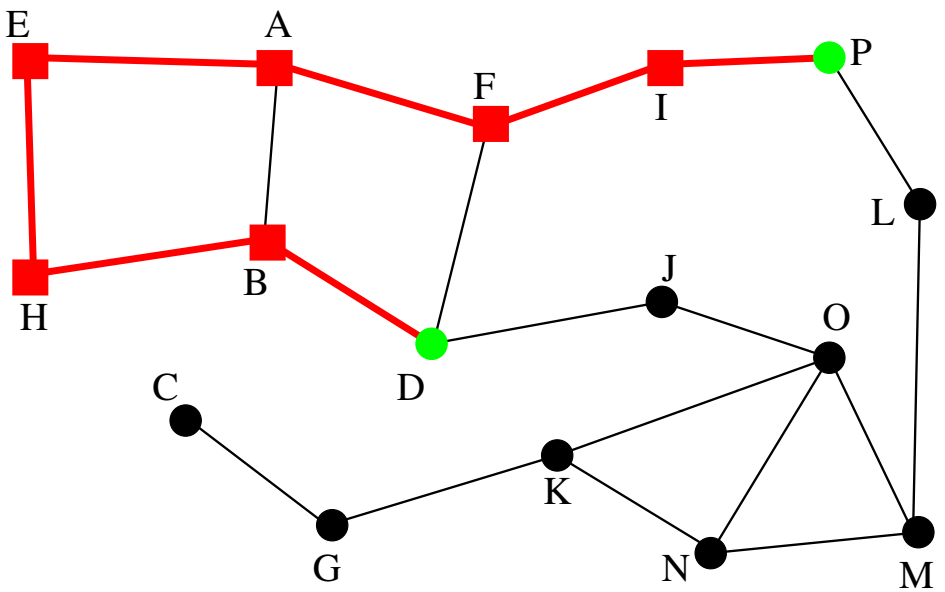
Parcours quelconque

▸ Fin Animation



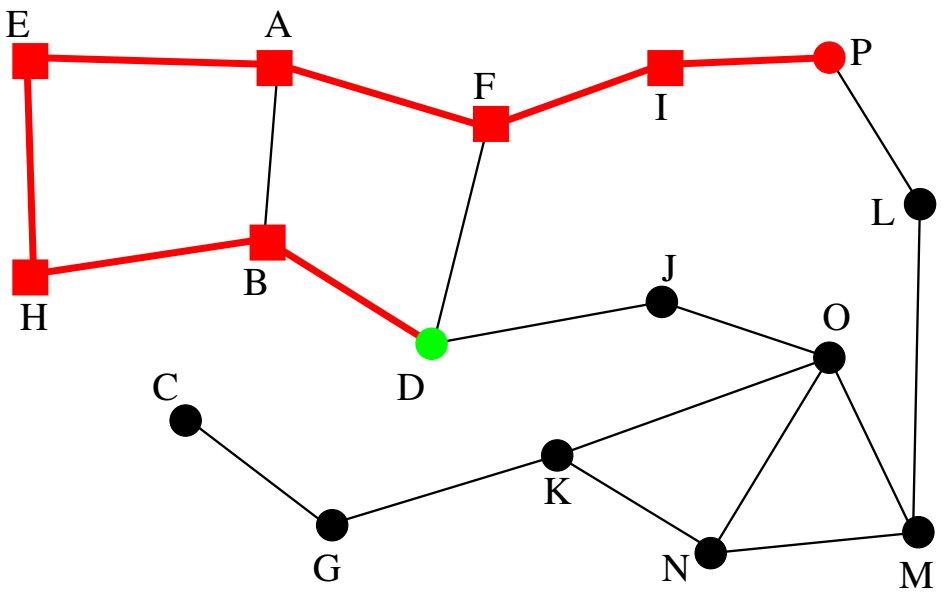
Parcours quelconque

▸ Fin Animation



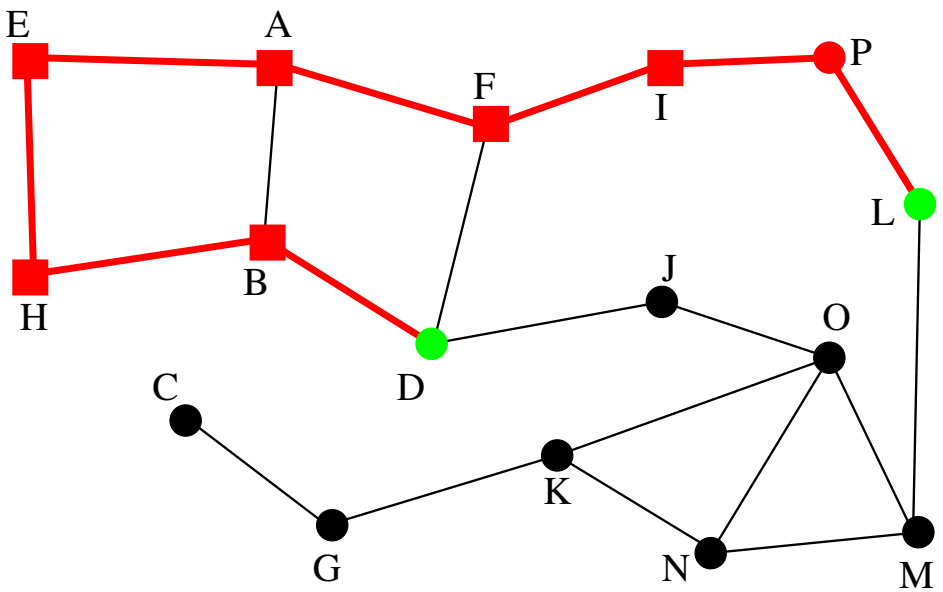
Parcours quelconque

▸ Fin Animation



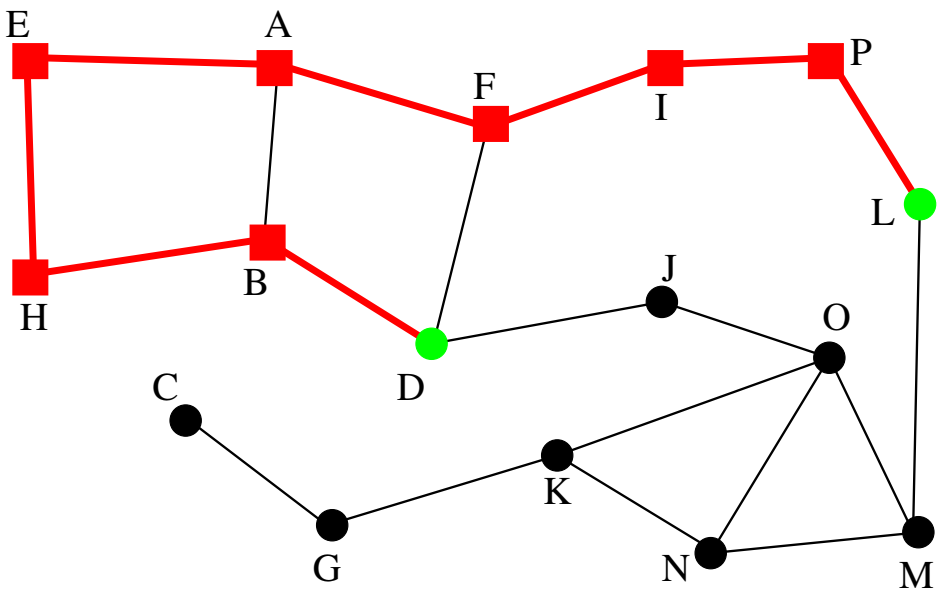
Parcours quelconque

▸ Fin Animation



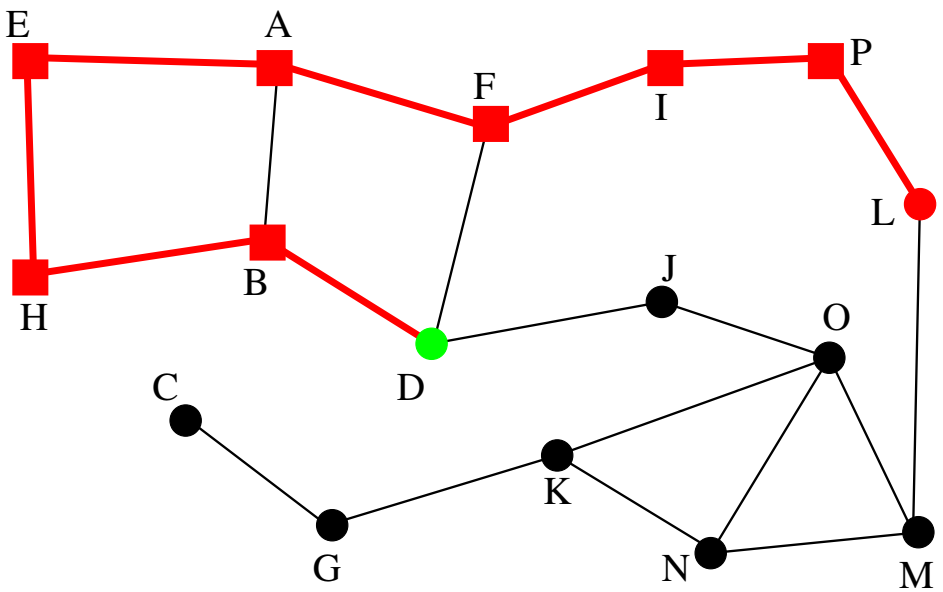
Parcours quelconque

▸ Fin Animation



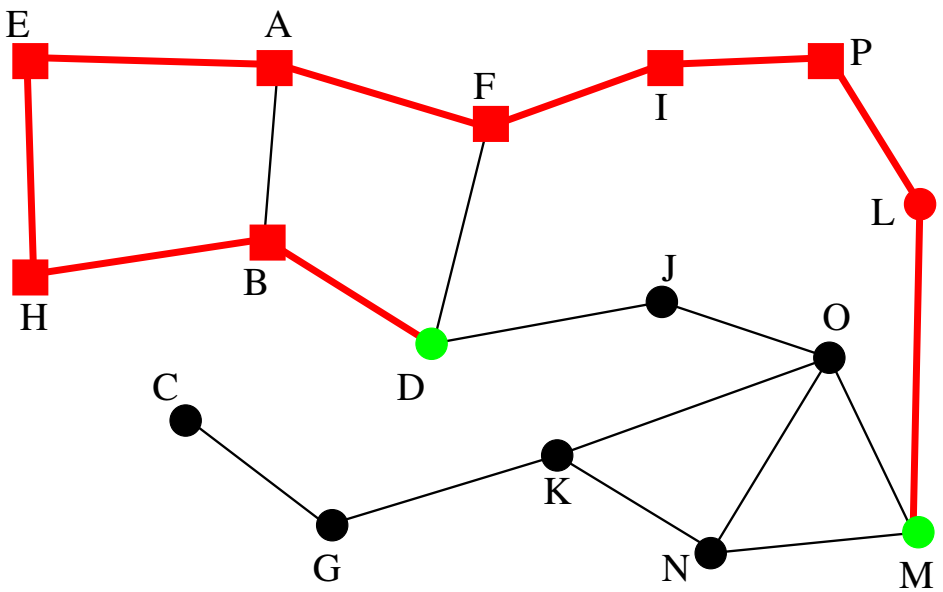
Parcours quelconque

▸ Fin Animation



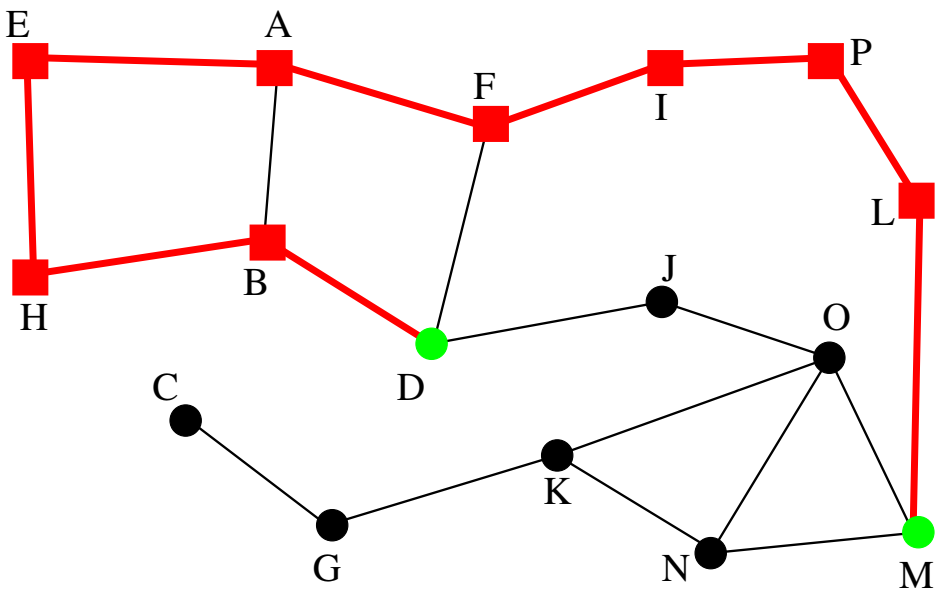
Parcours quelconque

▸ Fin Animation



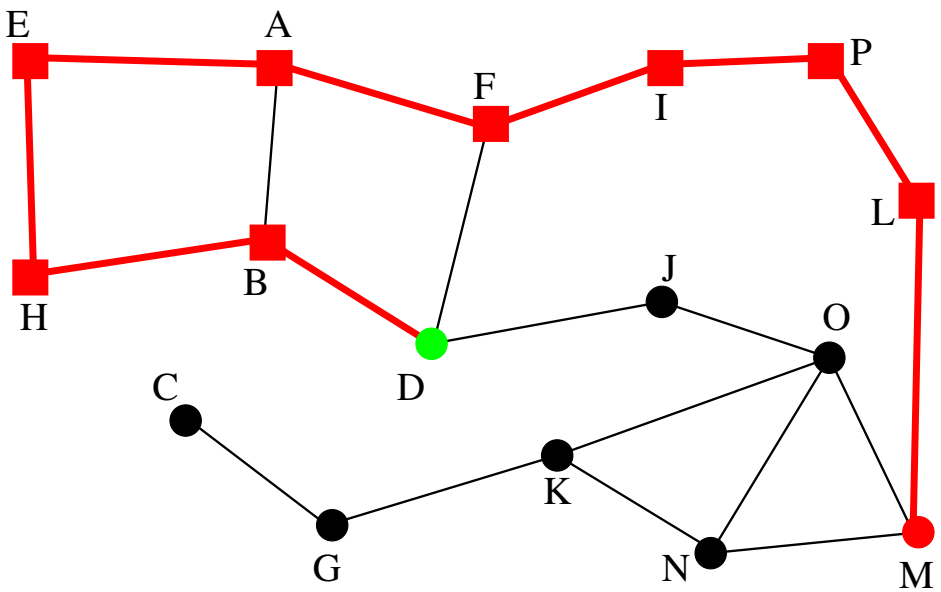
Parcours quelconque

▸ Fin Animation



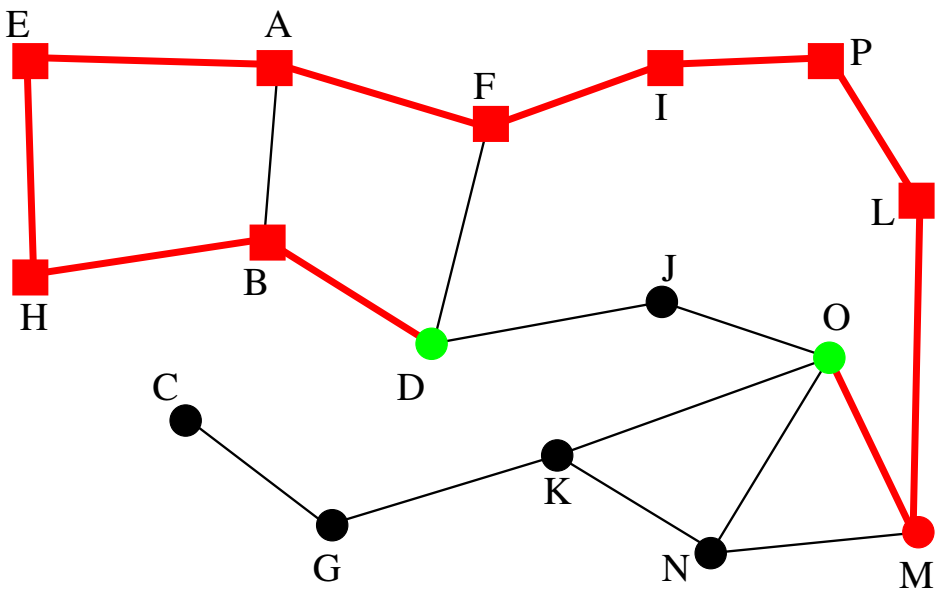
Parcours quelconque

▸ Fin Animation



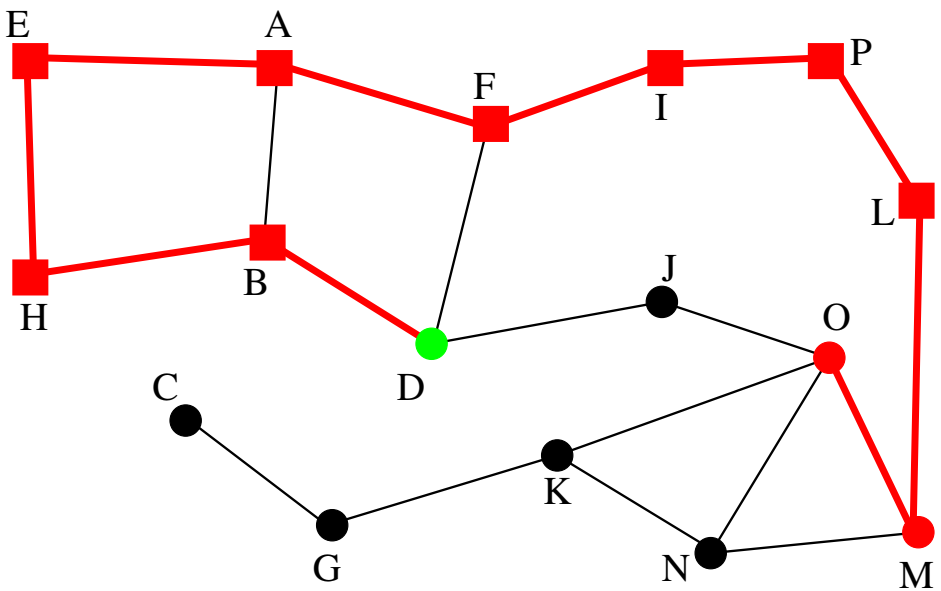
Parcours quelconque

▸ Fin Animation



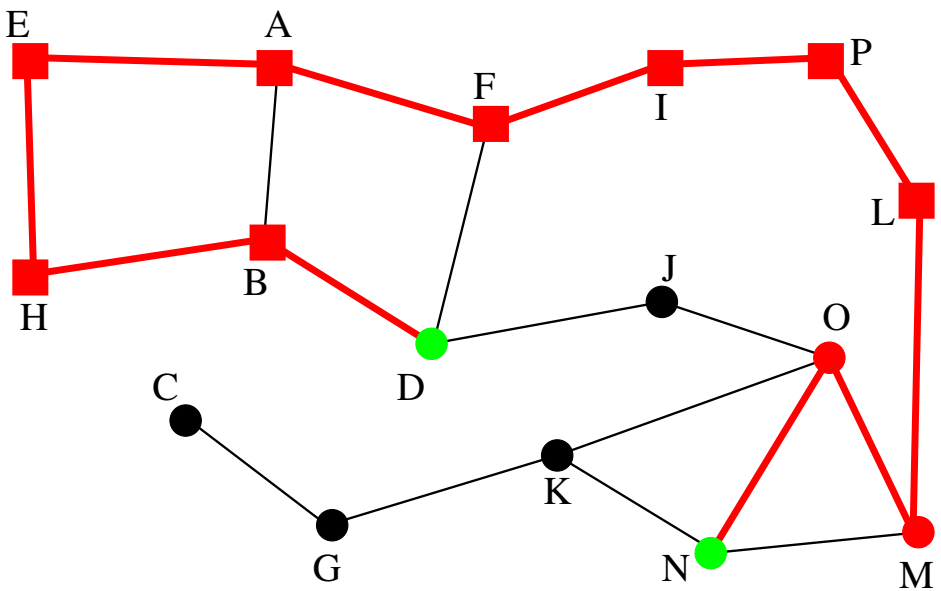
Parcours quelconque

▸ Fin Animation



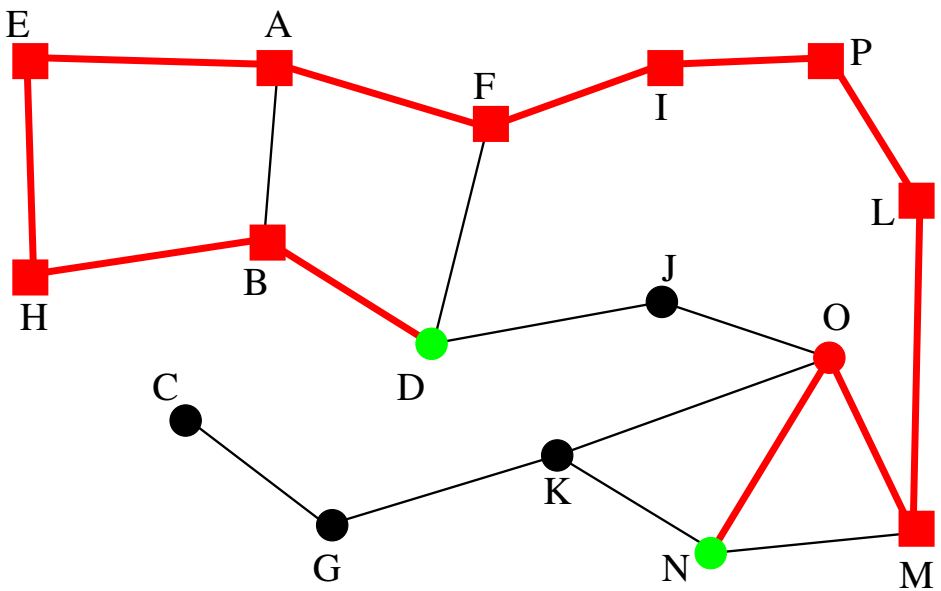
Parcours quelconque

▸ Fin Animation



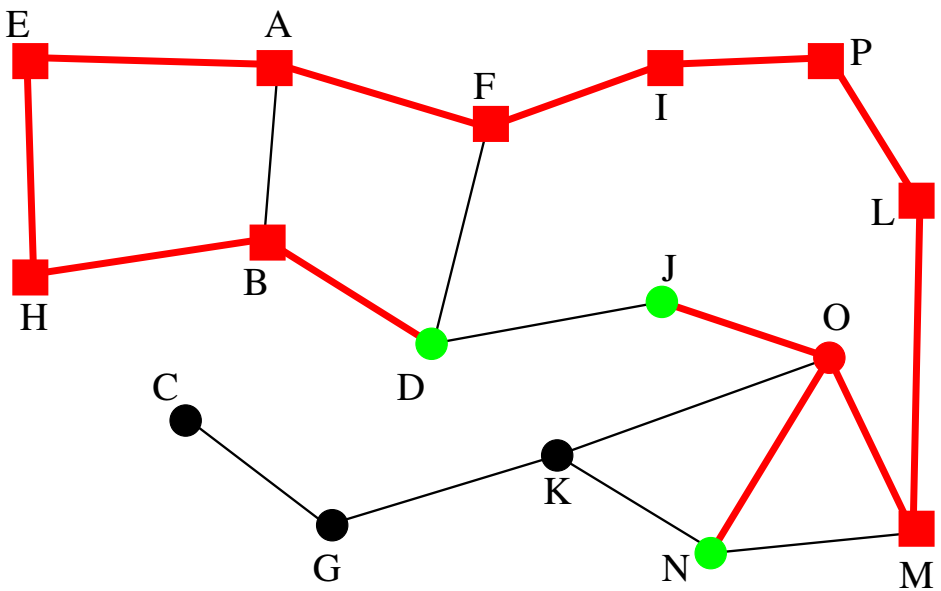
Parcours quelconque

▸ Fin Animation



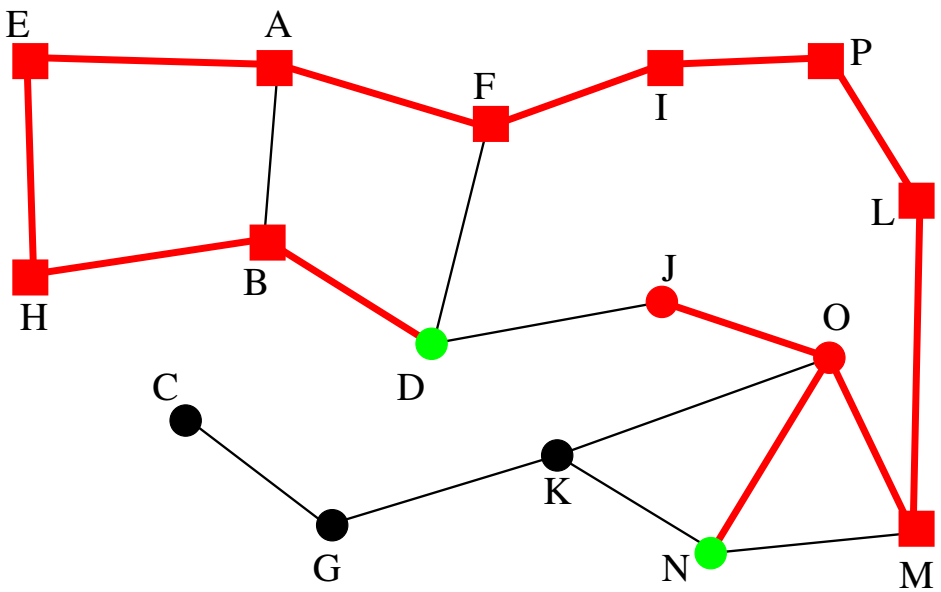
Parcours quelconque

▸ Fin Animation



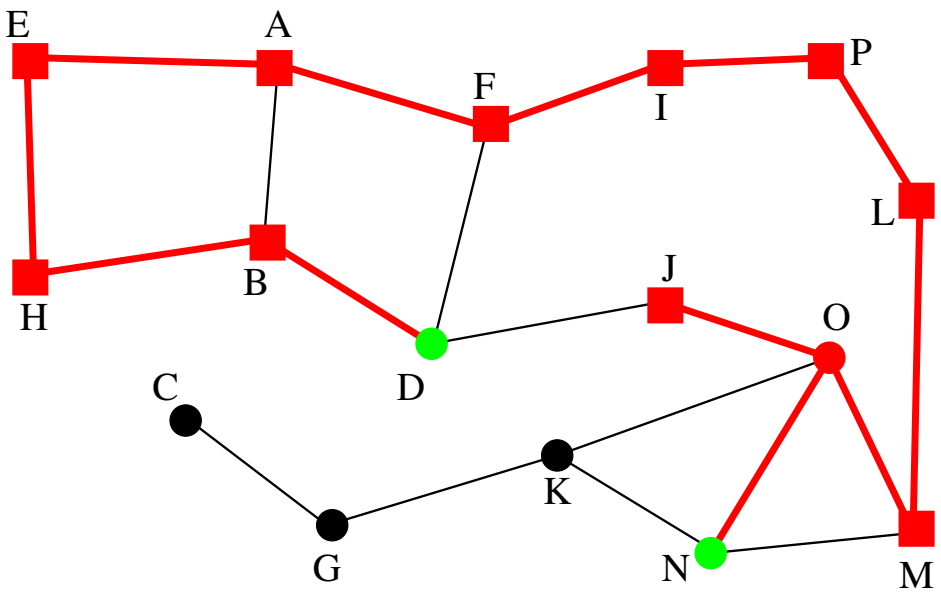
Parcours quelconque

▸ Fin Animation



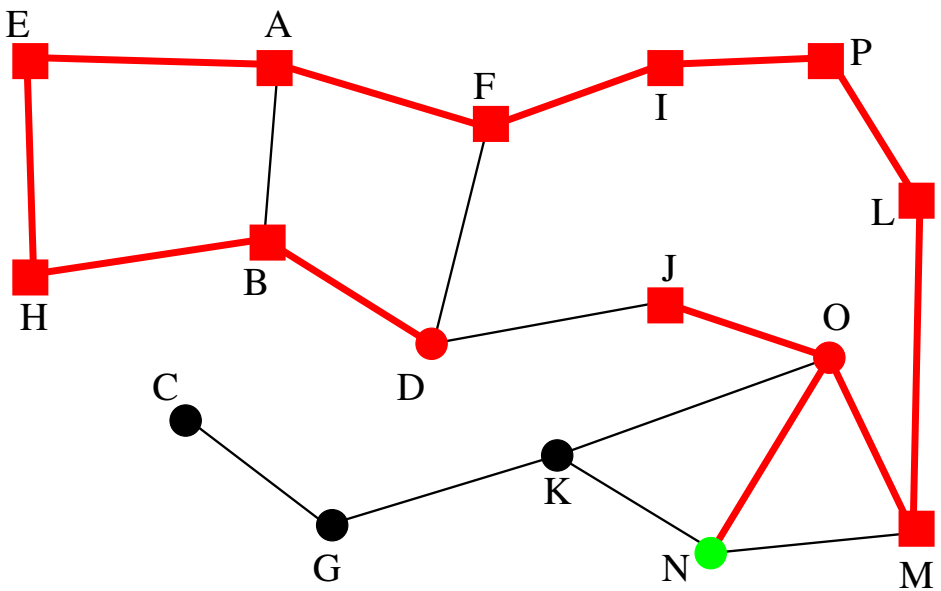
Parcours quelconque

▸ Fin Animation



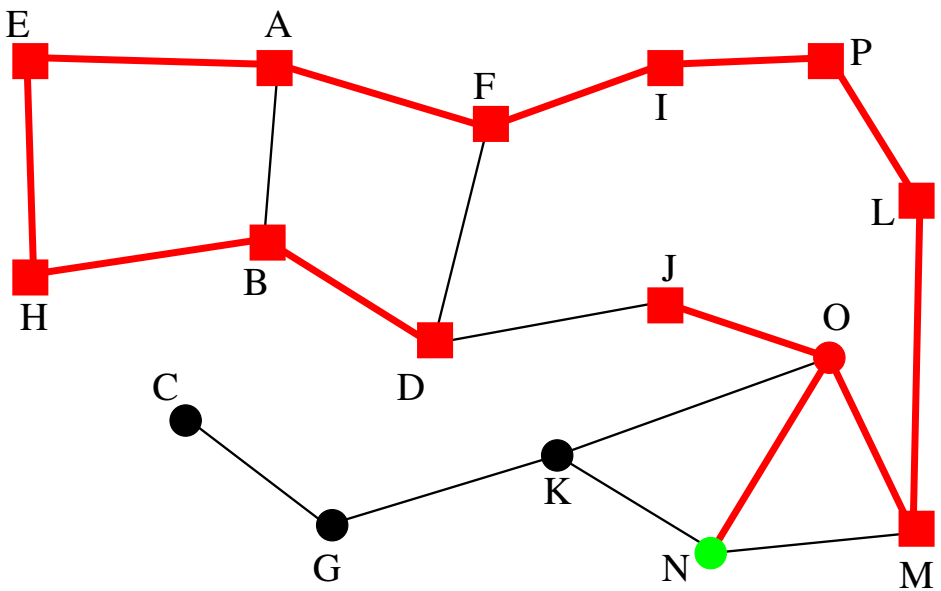
Parcours quelconque

▸ Fin Animation



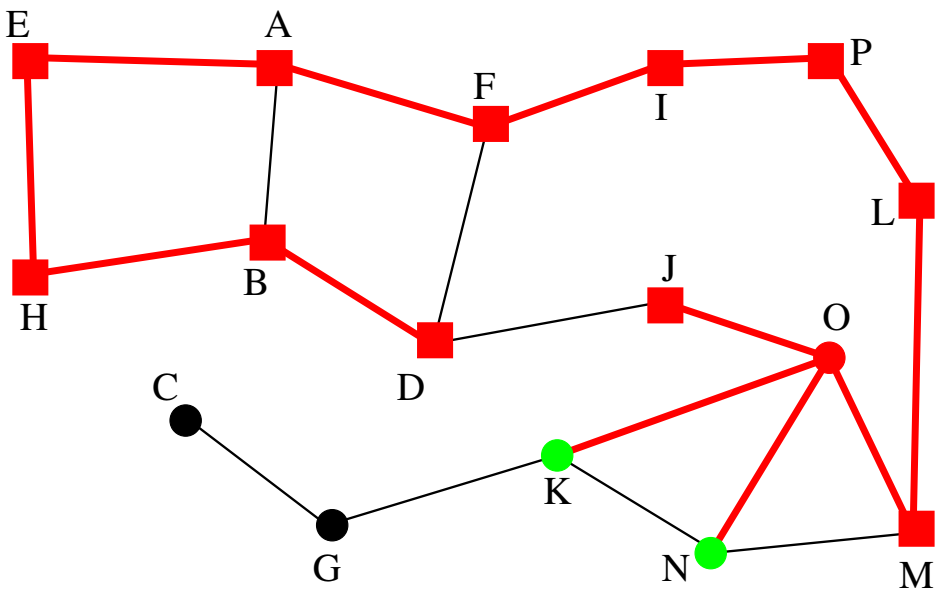
Parcours quelconque

▸ Fin Animation



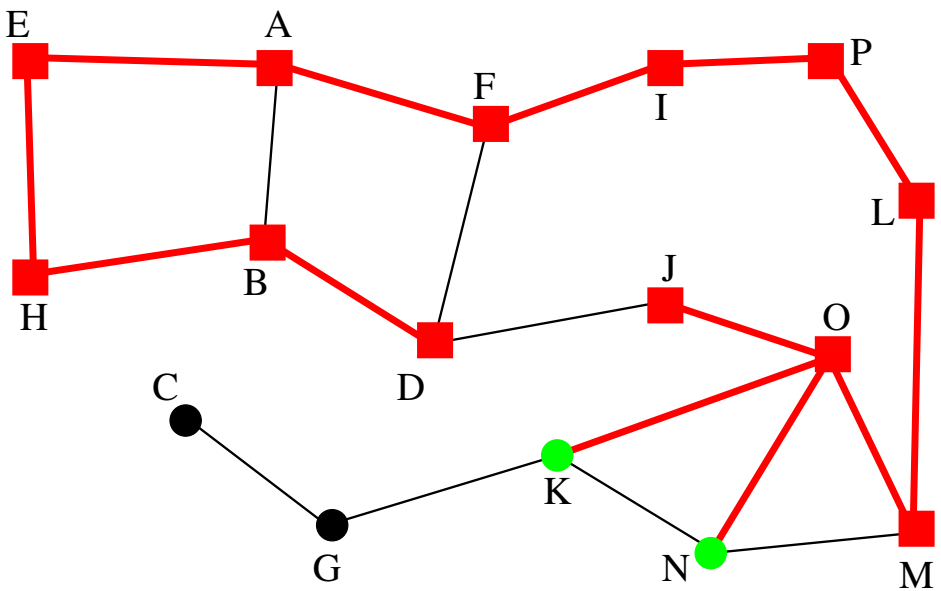
Parcours quelconque

▸ Fin Animation



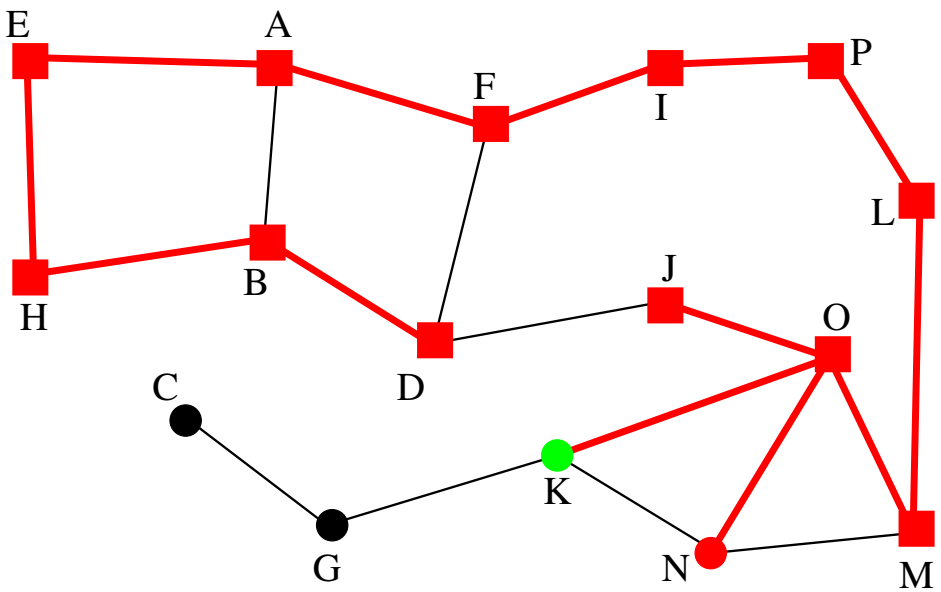
Parcours quelconque

▸ Fin Animation



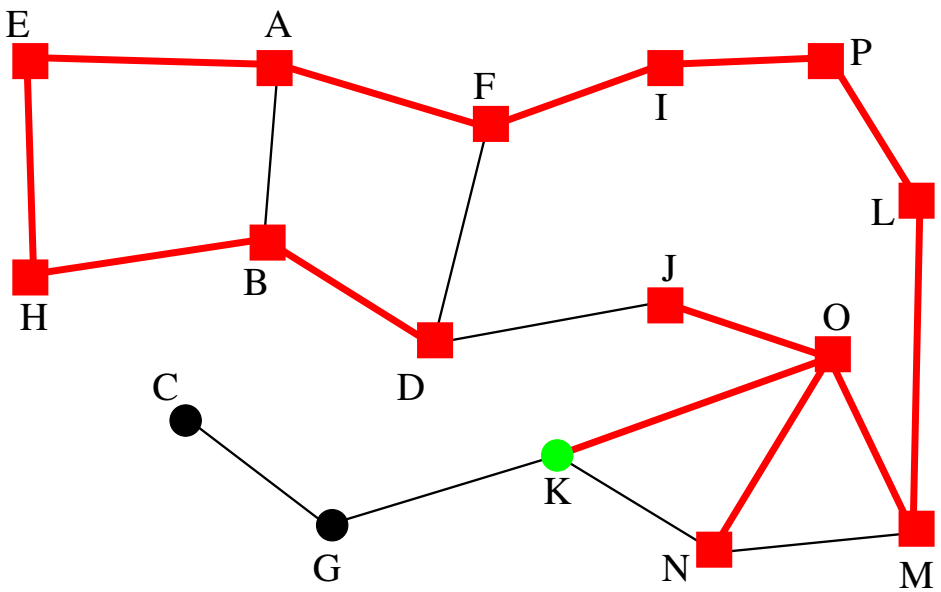
Parcours quelconque

▸ Fin Animation



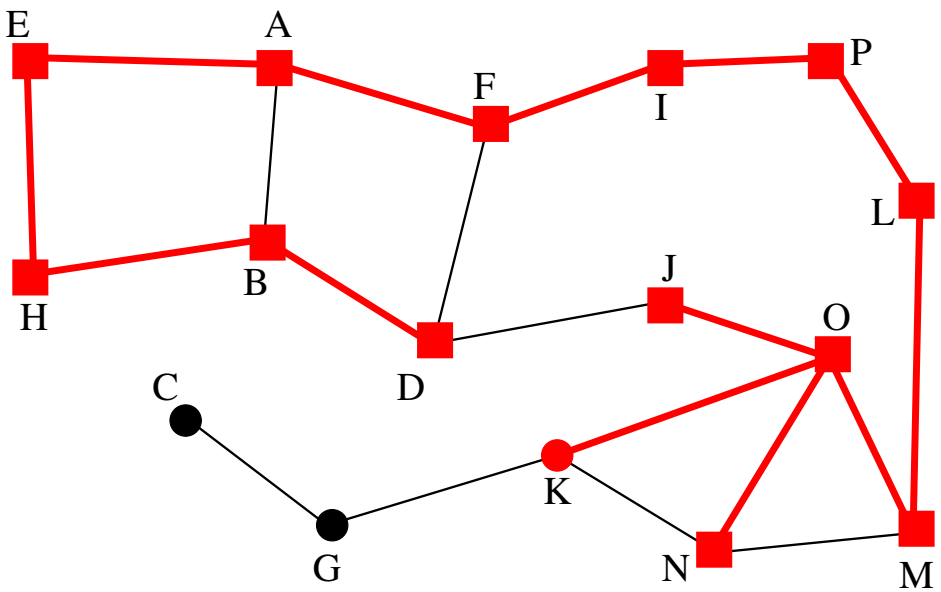
Parcours quelconque

▸ Fin Animation



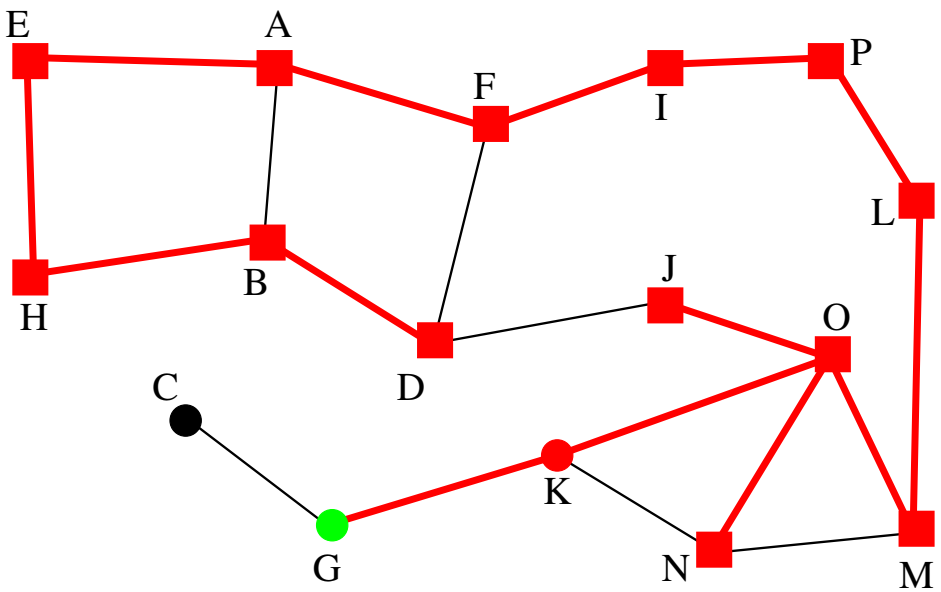
Parcours quelconque

▸ Fin Animation



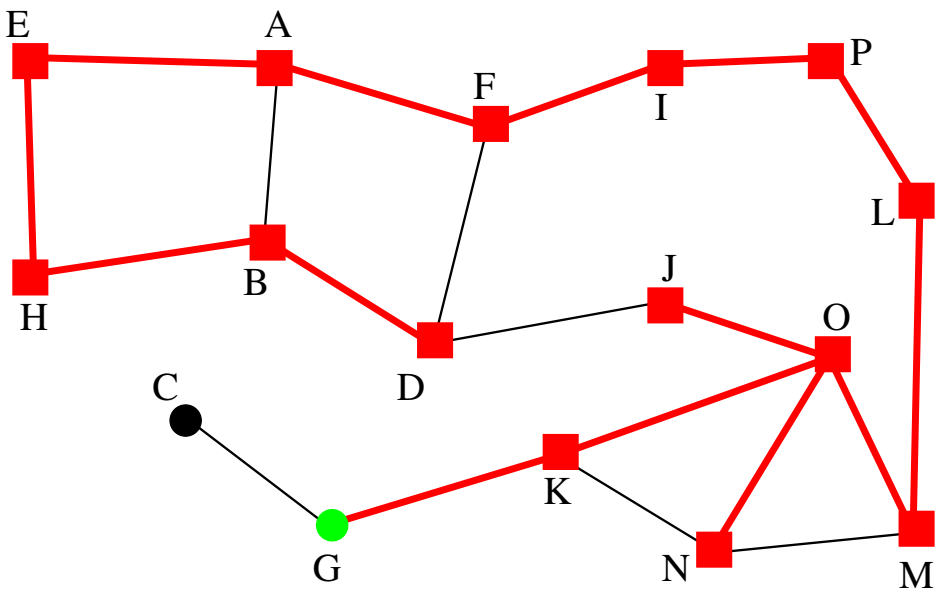
Parcours quelconque

▸ Fin Animation



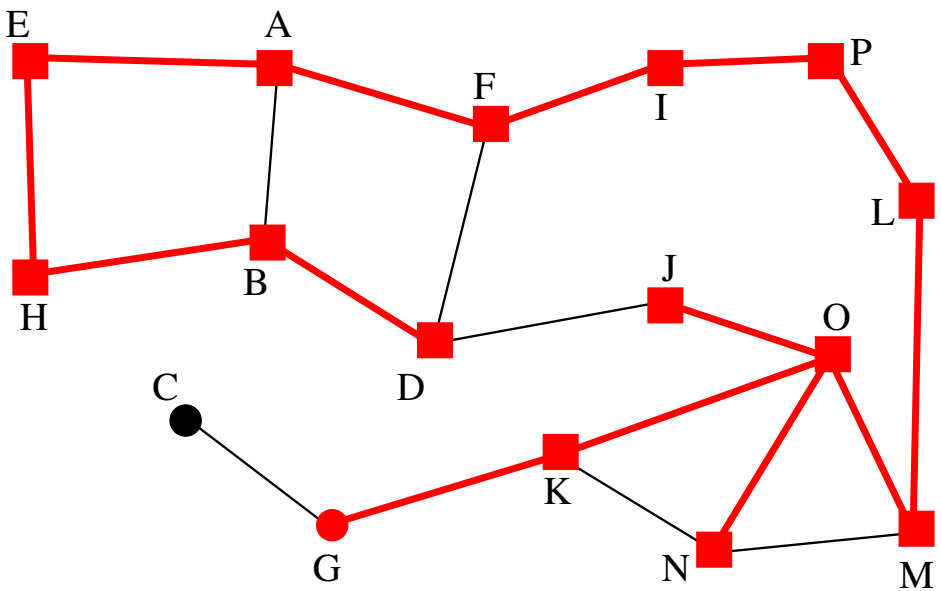
Parcours quelconque

▸ Fin Animation



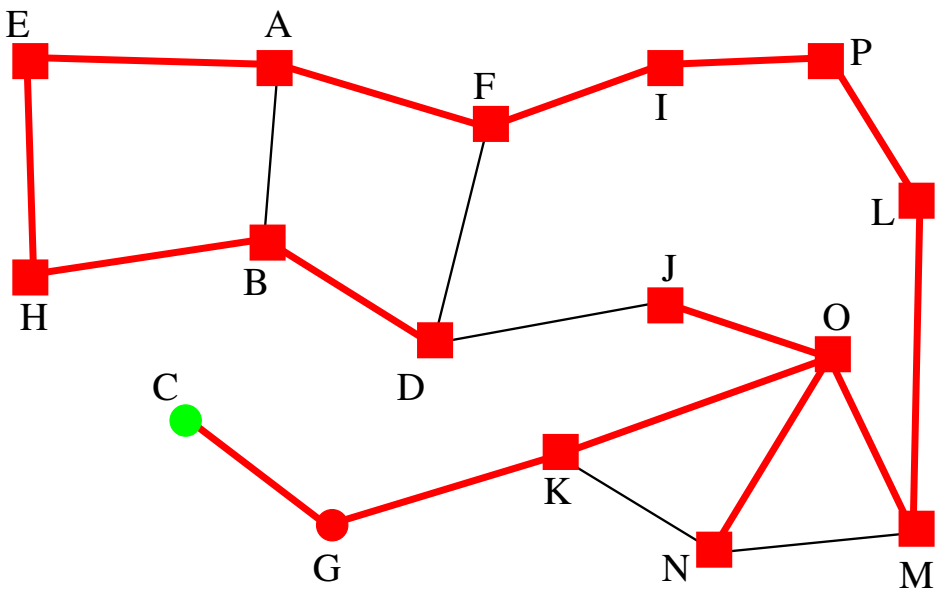
Parcours quelconque

▸ Fin Animation



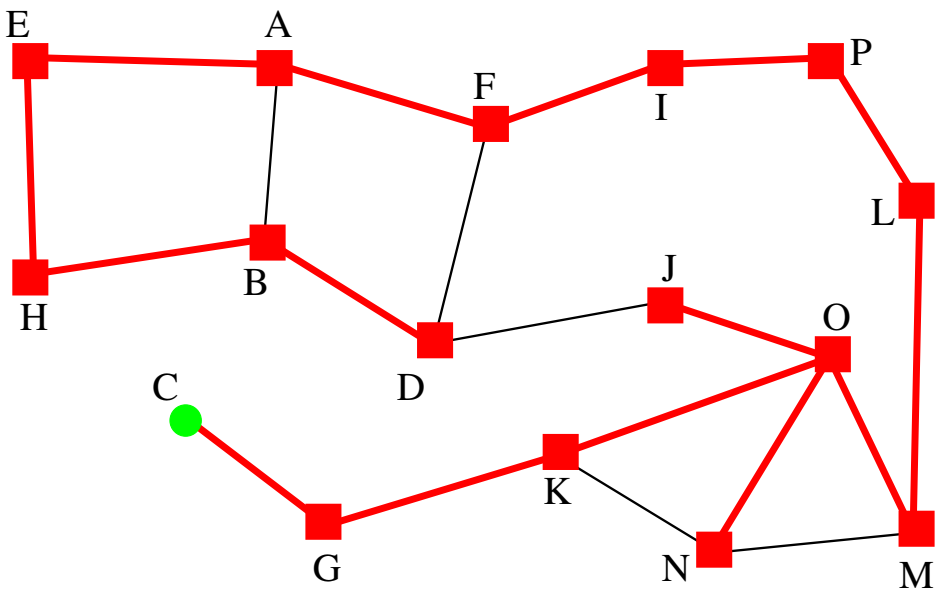
Parcours quelconque

▸ Fin Animation



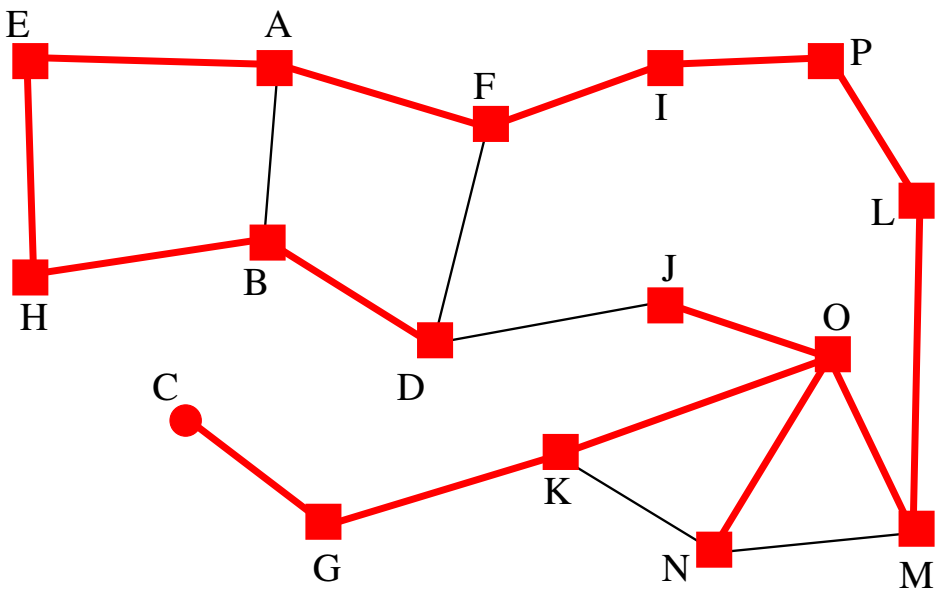
Parcours quelconque

▸ Fin Animation



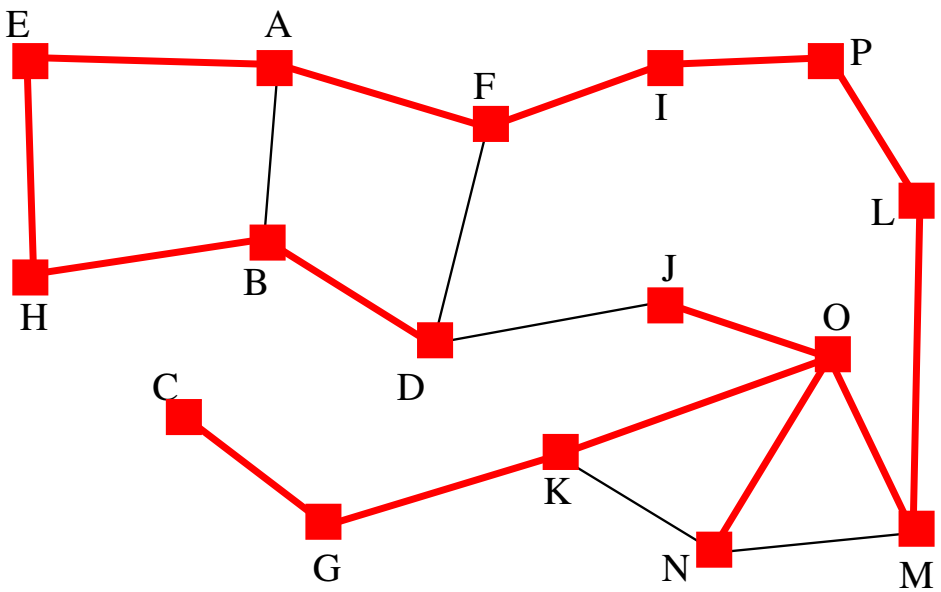
Parcours quelconque

▸ Fin Animation



Parcours quelconque

▸ Fin Animation



Comment organiser la frontière ?

Le plus simple : au hasard

À chaque nouvelle étape, on ajoute un sommet au hasard dans la frontière.

- Comme dans l'exemple précédent.
- On obtient alors un arbre couvrant quelconque.

Nous allons maintenant voir deux méthodes plus intelligentes d'organisation de la frontière.

Comment organiser la frontière ?

Le plus simple : au hasard

À chaque nouvelle étape, on ajoute un sommet au hasard dans la frontière.

- Comme dans l'exemple précédent.
- On obtient alors un arbre couvrant quelconque.

Nous allons maintenant voir deux méthodes plus intelligentes d'organisation de la frontière.

Sommaire

1. Introduction
2. Exemple
3. File et pile
4. Parcours en largeur
   - Rappel sur les files
   - Algorithme
   - Arbre de parcours et calcul des distances
5. Parcours en profondeur
   - Rappel sur les piles
   - Algorithme
   - Arbre de parcours
6. Pour finir

Comment organiser la frontière ?

Méthode 1 : Premier arrivé = premier servi

On fait attendre un sommet le moins possible dans la frontière, on traite la frontière comme une file d'attente.

- On obtient un arbre tel que le niveau d'un sommet dans l'arbre correspond à la distance au sommet de départ.
- On parle de parcours en largeur.

Édouard Manet *La queue devant la Boucherie*

Comment organiser la frontière ?

Méthode 2 : Dernier arrivé = premier servi

Ou bien au contraire, on traite la frontière comme une pile.

- On obtient un arbre tel que toute arête du graphe qui n'est pas dans l'arbre va forcément depuis un sommet vers un de ses ancêtres.
- On parle de parcours en profondeur.

En résumé

- Parcourir un graphe connexe induit un arbre couvrant.
- Différentes méthodes selon la structure de données choisie pour stocker la frontière.
  - File d'attente
    - Parcours en largeur
    - Distance à la source
  - Pile
    - Parcours en profondeur
    - Tri topologique (sera vu plus tard)

Sommaire

1 Introduction

2 Exemple

3 File et pile

4 Parcours en largeur
- Rappel sur les files
- Algorithme
- Arbre de parcours et calcul des distances

5 Parcours en profondeur
- Rappel sur les piles
- Algorithme
- Arbre de parcours

6 Pour finir

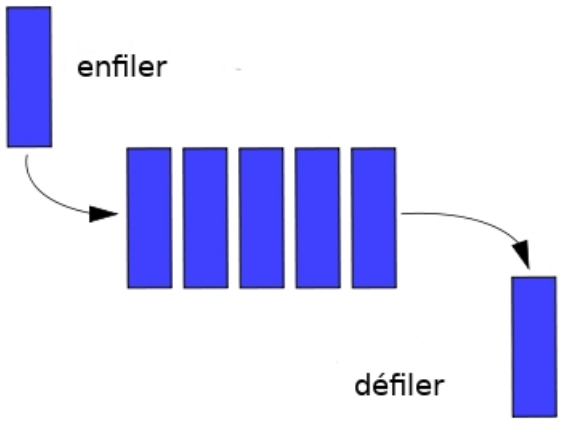
Sommaire

1. Introduction
2. Exemple
3. File et pile
4. Parcours en largeur
   - Rappel sur les files
   - Algorithme
   - Arbre de parcours et calcul des distances
5. Parcours en profondeur
   - Rappel sur les piles
   - Algorithme
   - Arbre de parcours
6. Pour finir

Définition d'une file

Les primitives permettant de gérer une file sont :

- Ajouter un sommet (toujours en queue de file) (`enfiler`)
- Enlever un sommet (toujours en tête de file) (`défiler`)
- Vérifier si la file est vide (`estVide?`)



- On peut aussi juste regarder le sommet en tête de file sans l'enlever (`valeurTete`)

Les files en pratique

- En anglais : FIFO (*First In First Out*)
- Peut se coder facilement avec des listes chaînées (par exemple)
- Des librairies implémentant les files existent dans la majeure partie des langages de programmation

Exemple

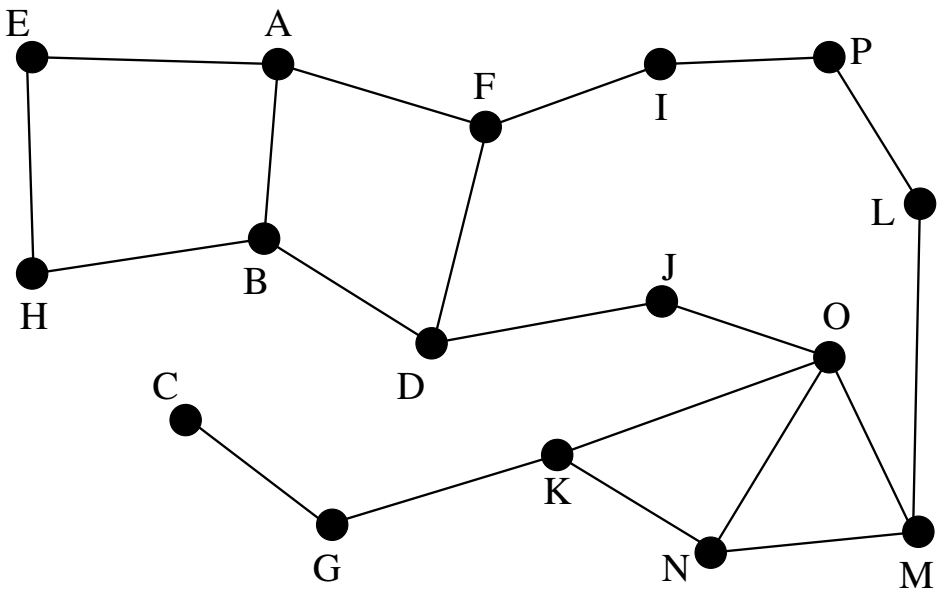
File d'attente associée à une imprimante

Sommaire

1. Introduction
2. Exemple
3. File et pile
4. Parcours en largeur
   - Rappel sur les files
   - Algorithme
   - Arbre de parcours et calcul des distances
5. Parcours en profondeur
   - Rappel sur les piles
   - Algorithme
   - Arbre de parcours
6. Pour finir

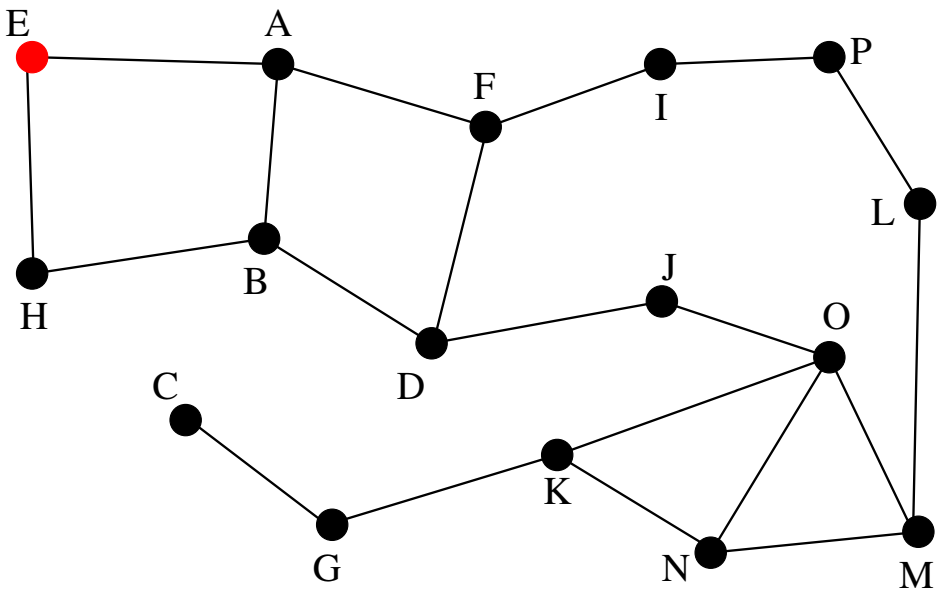
Exemple (parcours en largeur)

▸ Fin Animation



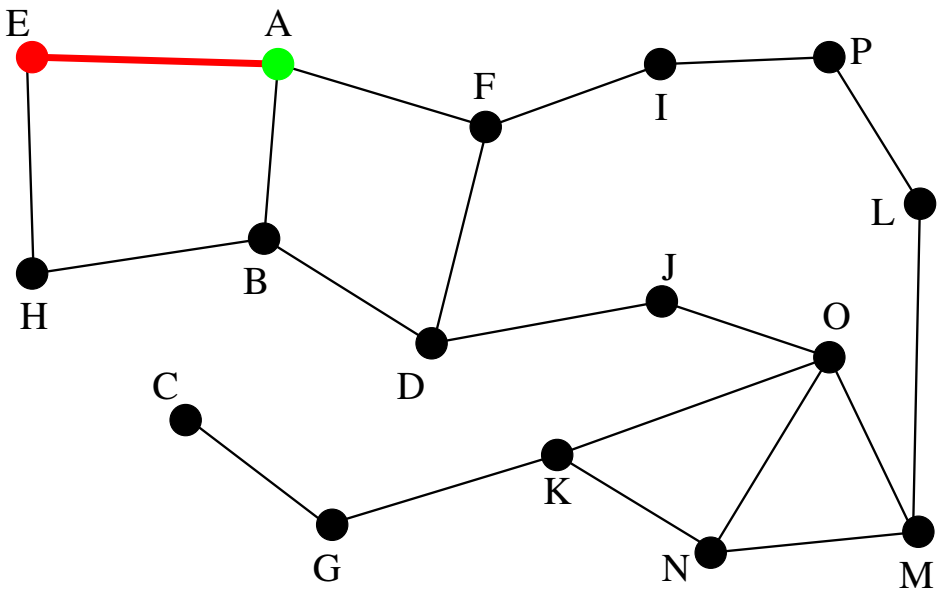
Exemple (parcours en largeur)

▸ Fin Animation



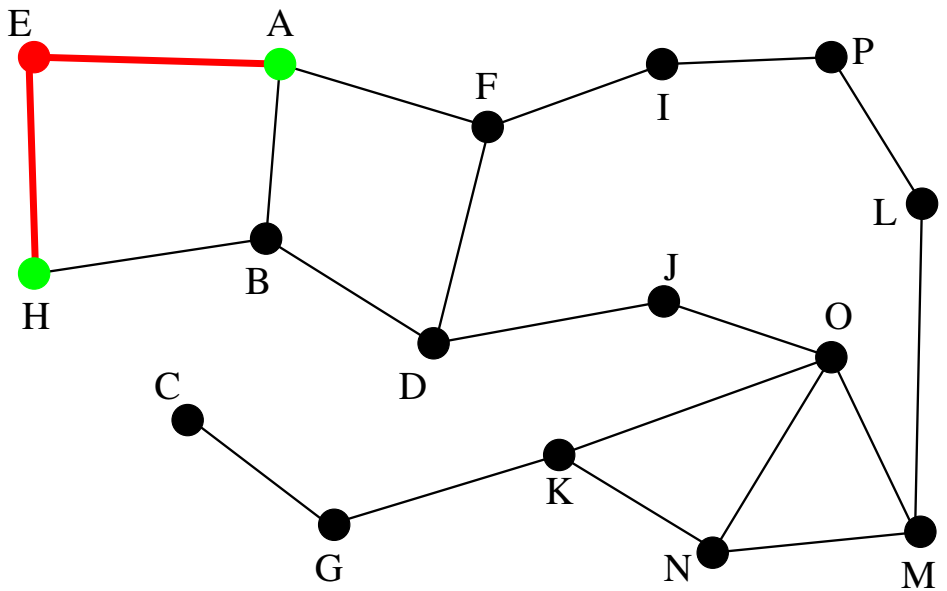
Exemple (parcours en largeur)

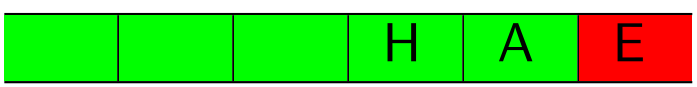
▸ Fin Animation



Exemple (parcours en largeur)

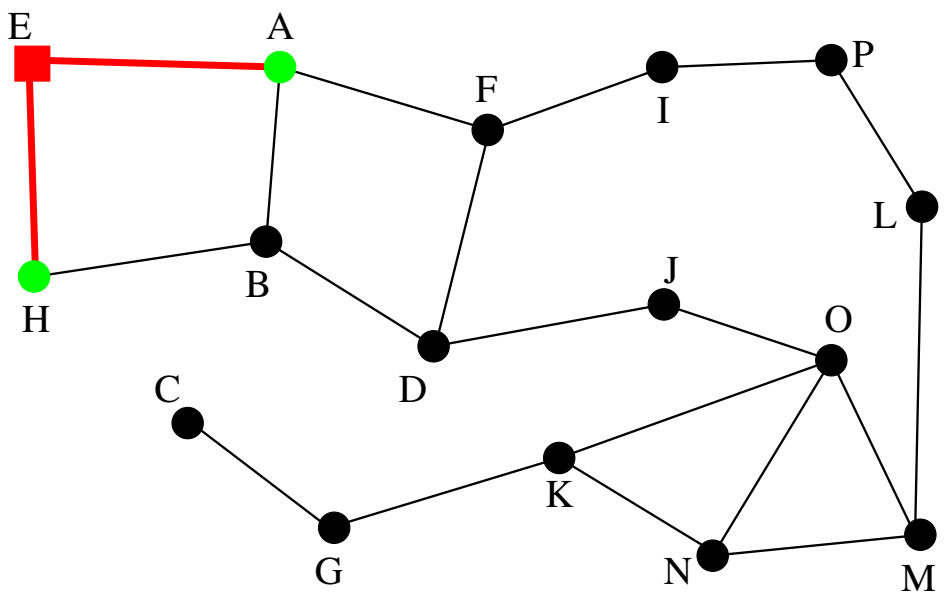
▸ Fin Animation

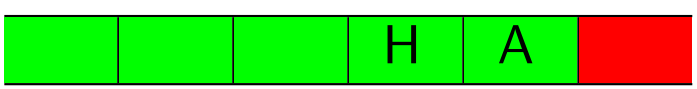




Exemple (parcours en largeur)

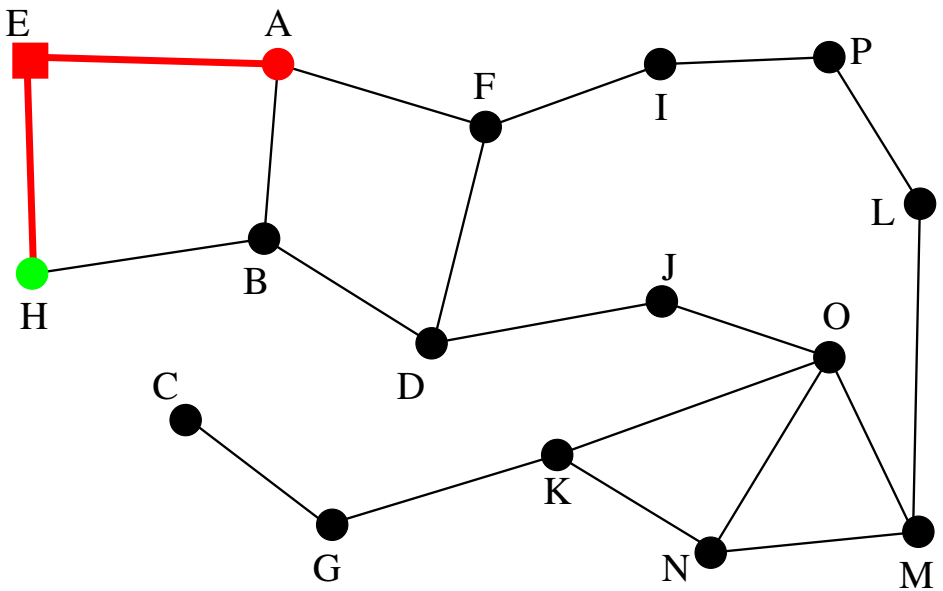
▸ Fin Animation





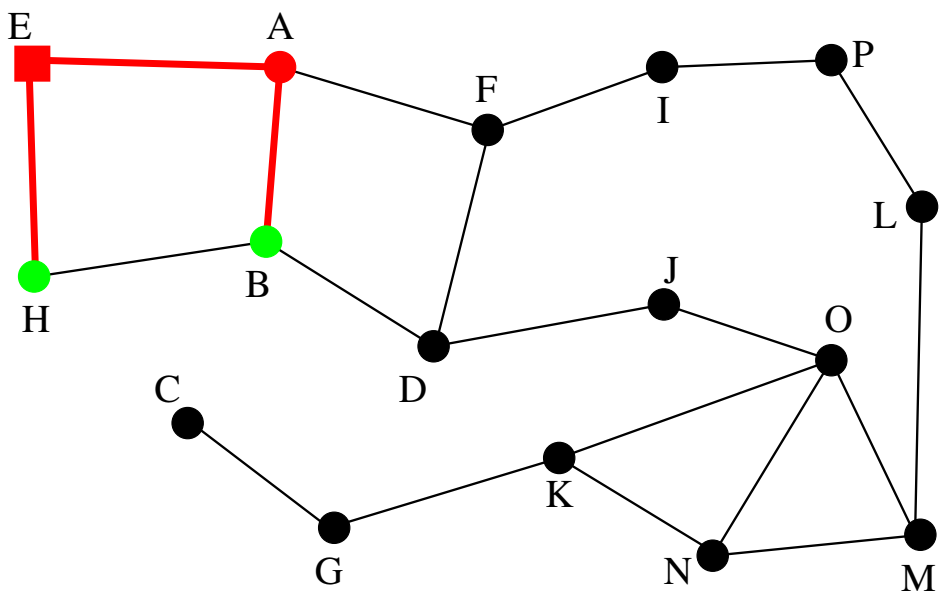
Exemple (parcours en largeur)

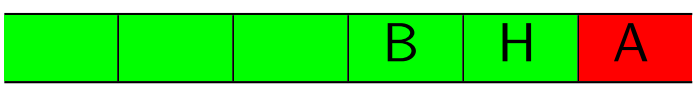
▸ Fin Animation



Exemple (parcours en largeur)

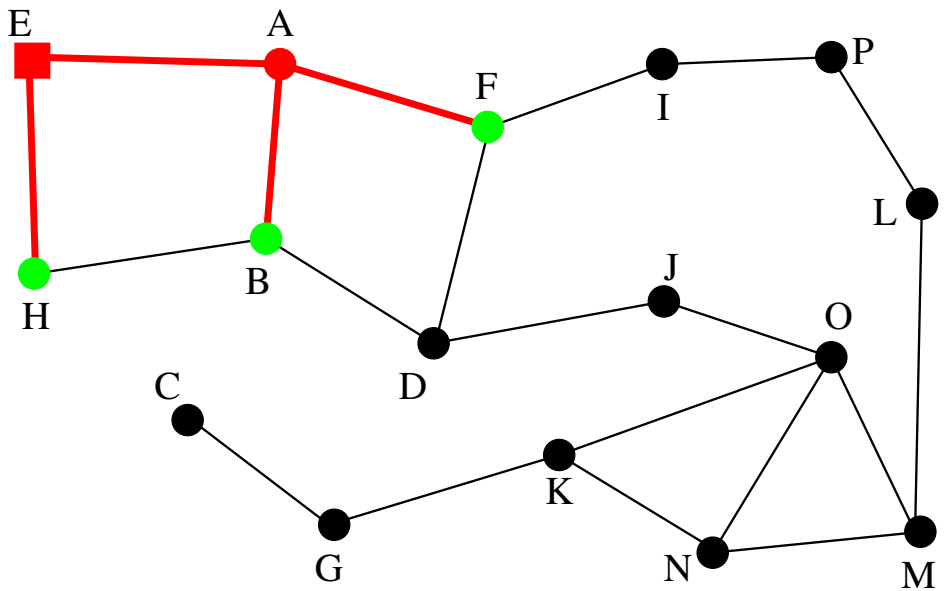
▸ Fin Animation

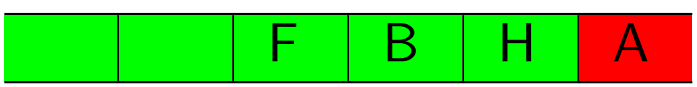




Exemple (parcours en largeur)

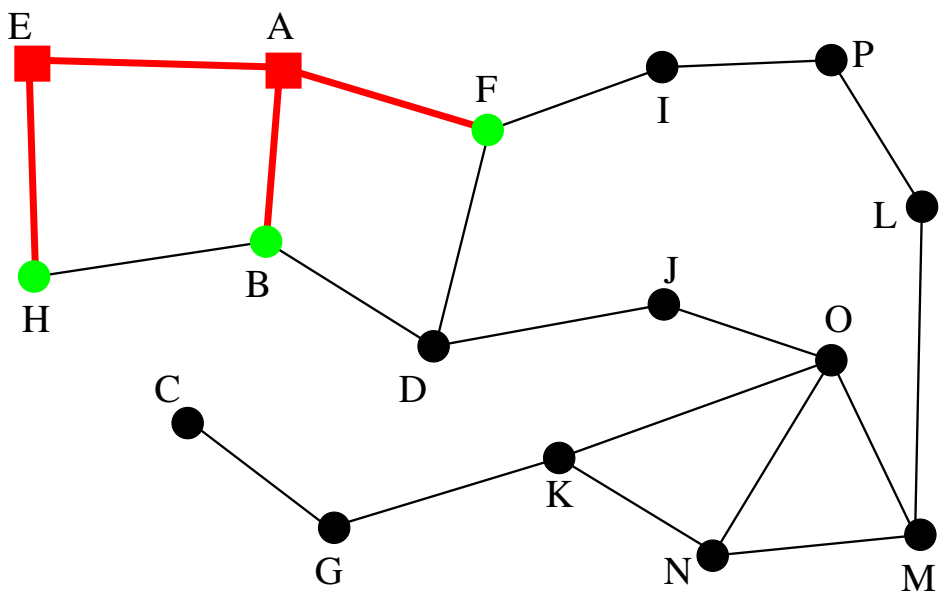
▸ Fin Animation

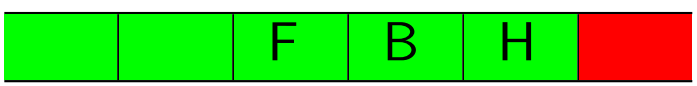




Exemple (parcours en largeur)

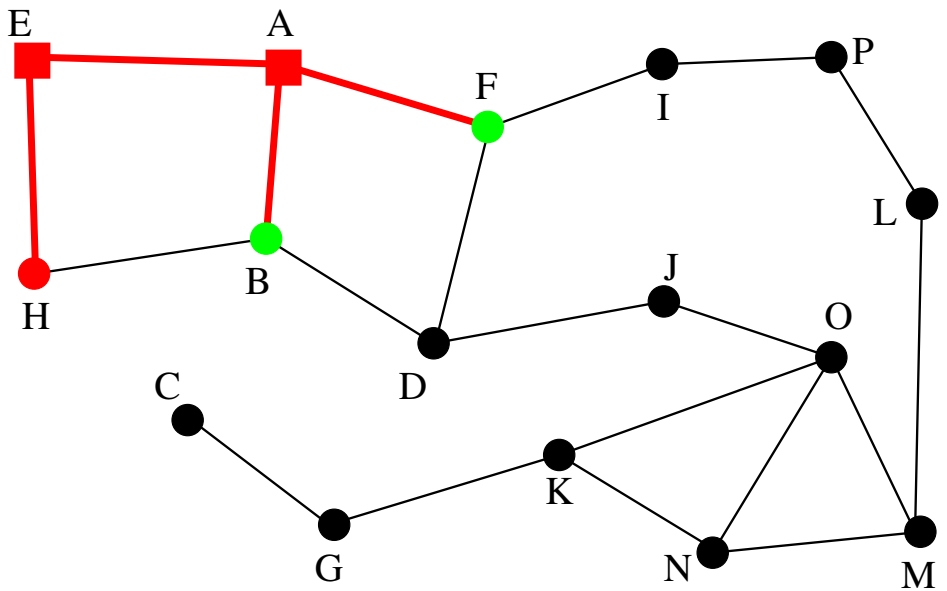
▸ Fin Animation

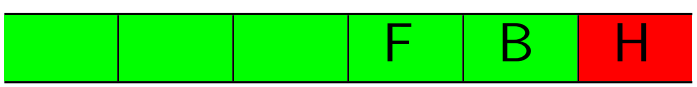




Exemple (parcours en largeur)

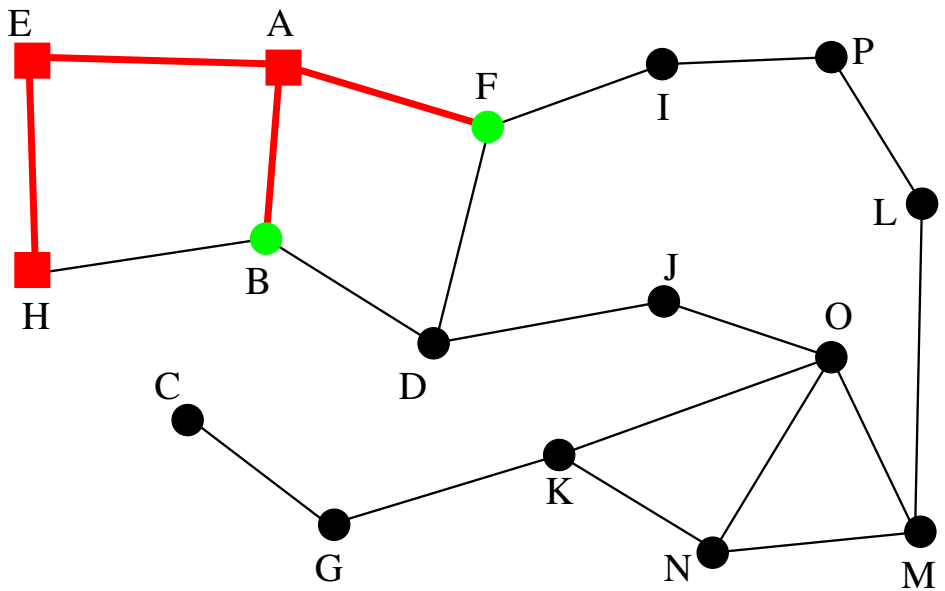
▸ Fin Animation

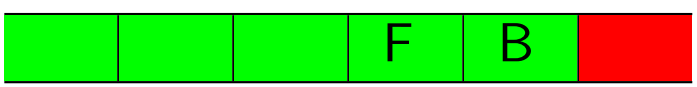




Exemple (parcours en largeur)

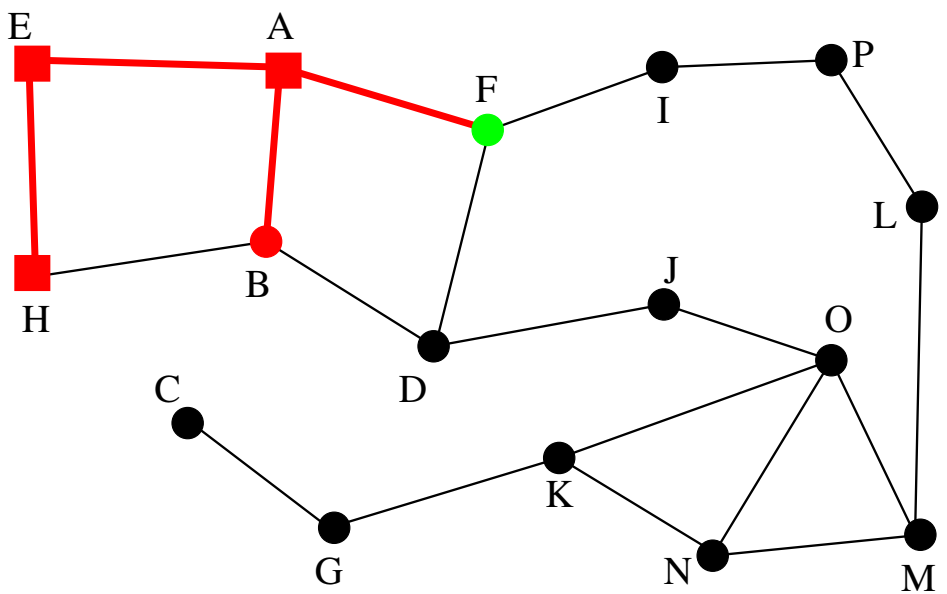
▸ Fin Animation





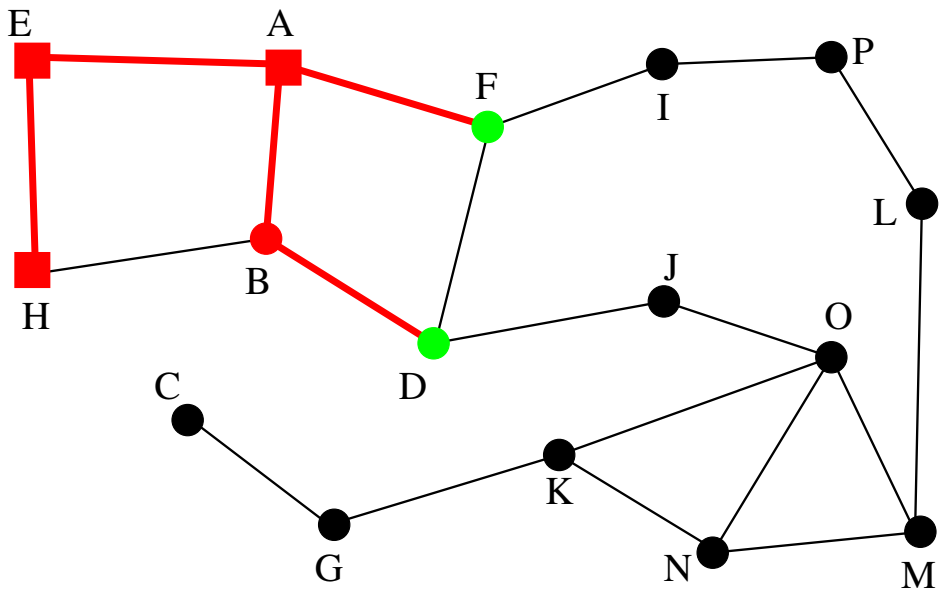
Exemple (parcours en largeur)

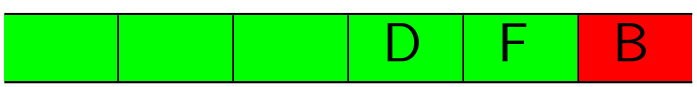
▸ Fin Animation



Exemple (parcours en largeur)

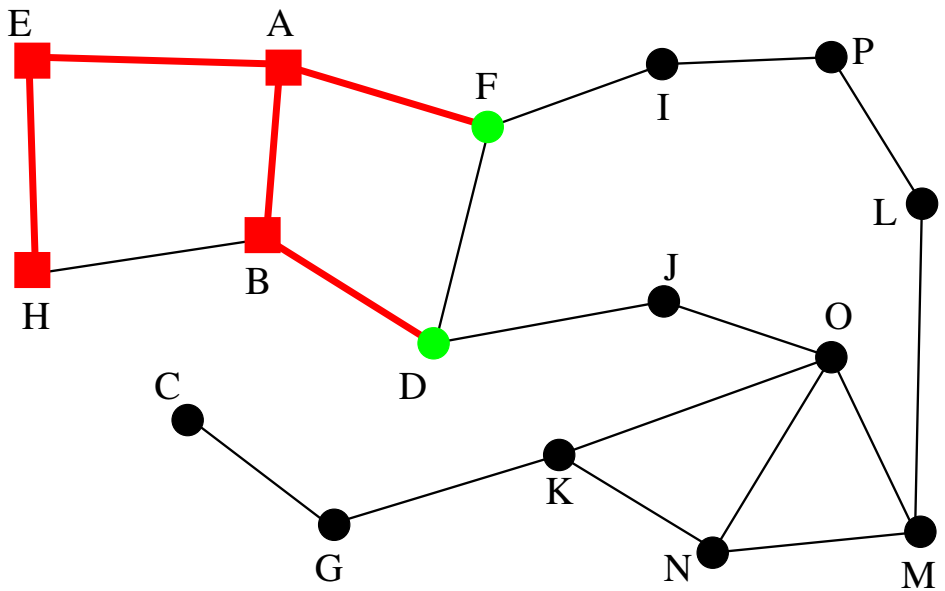
▸ Fin Animation

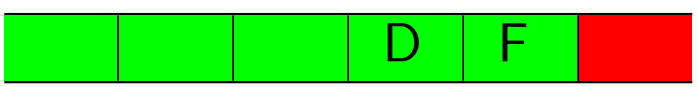




Exemple (parcours en largeur)

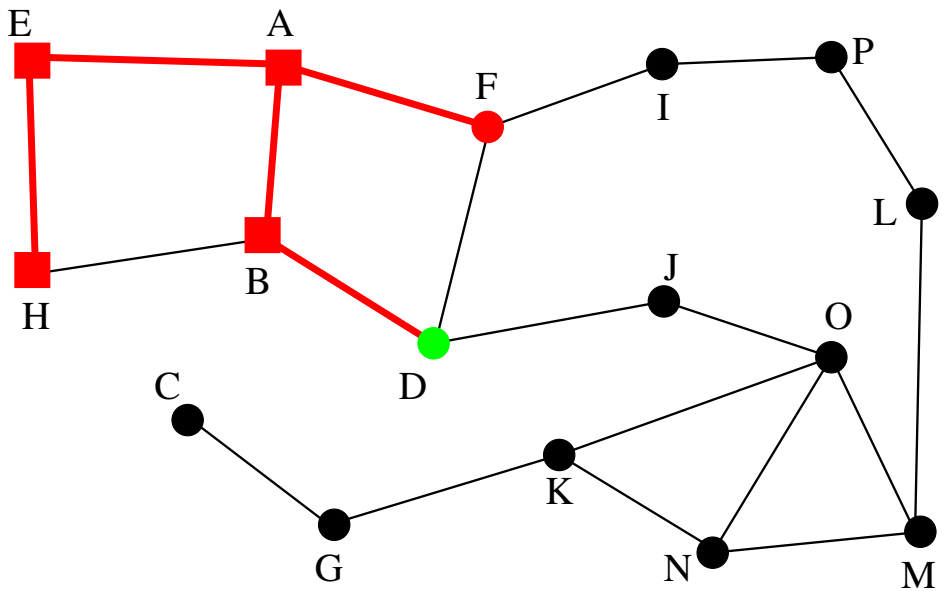
▸ Fin Animation





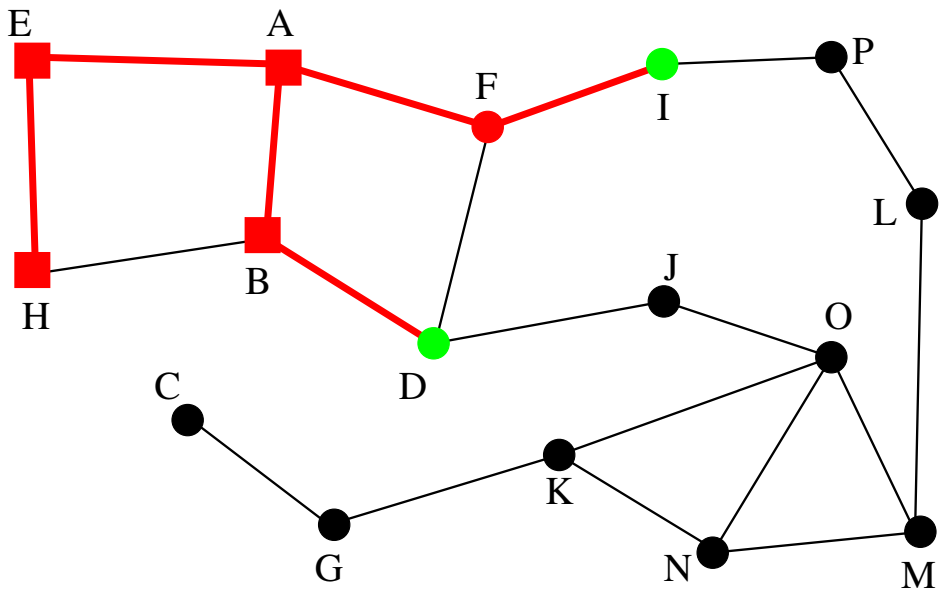
Exemple (parcours en largeur)

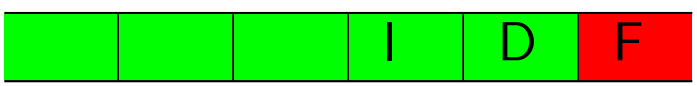
▸ Fin Animation



Exemple (parcours en largeur)

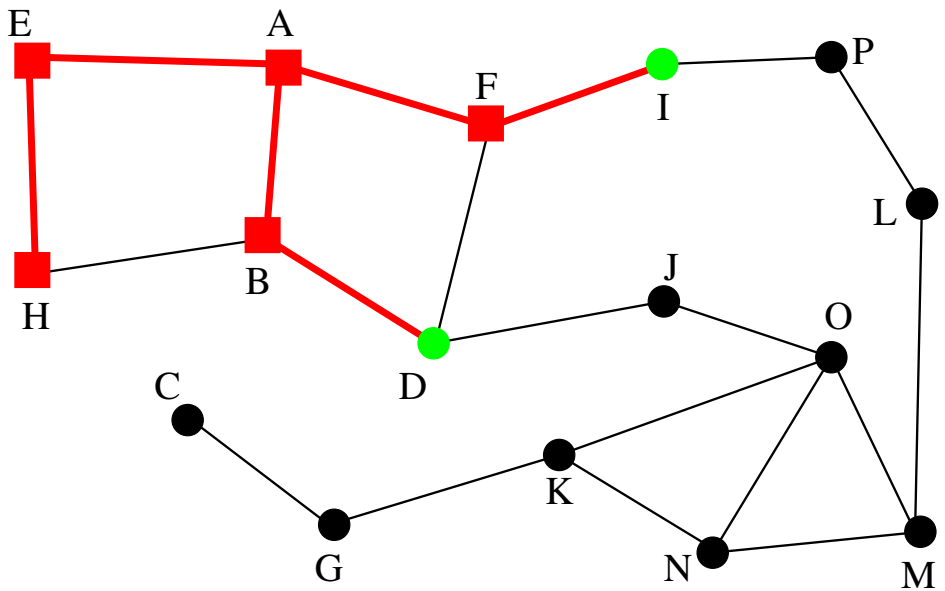
▸ Fin Animation

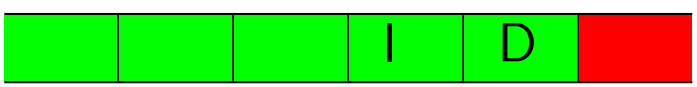




Exemple (parcours en largeur)

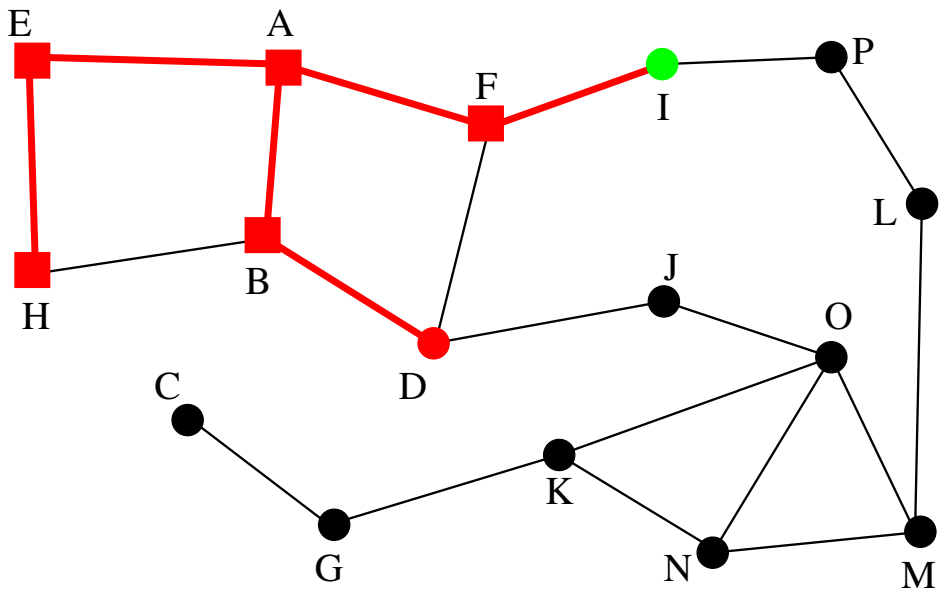
▸ Fin Animation





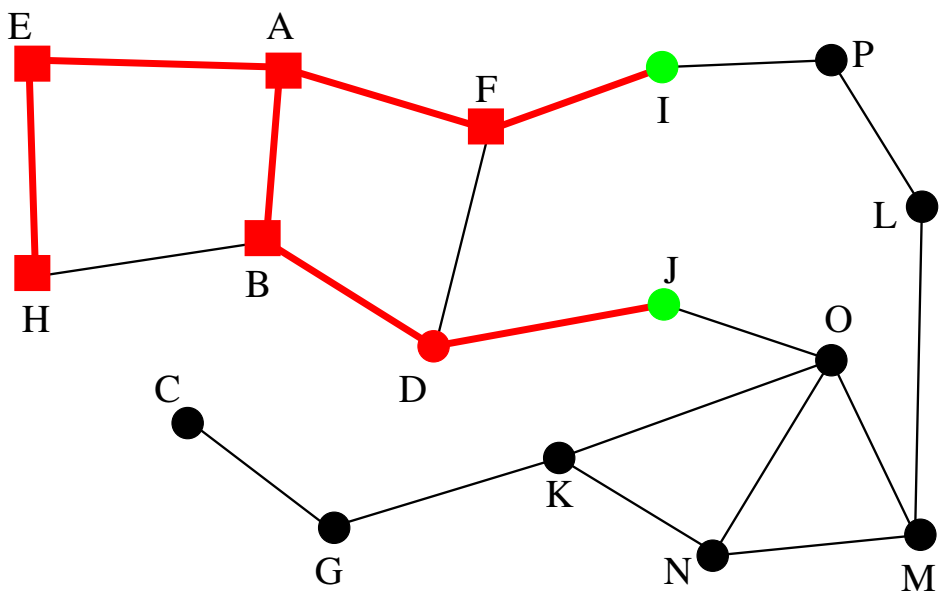
Exemple (parcours en largeur)

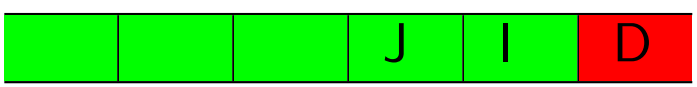
▸ Fin Animation



Exemple (parcours en largeur)

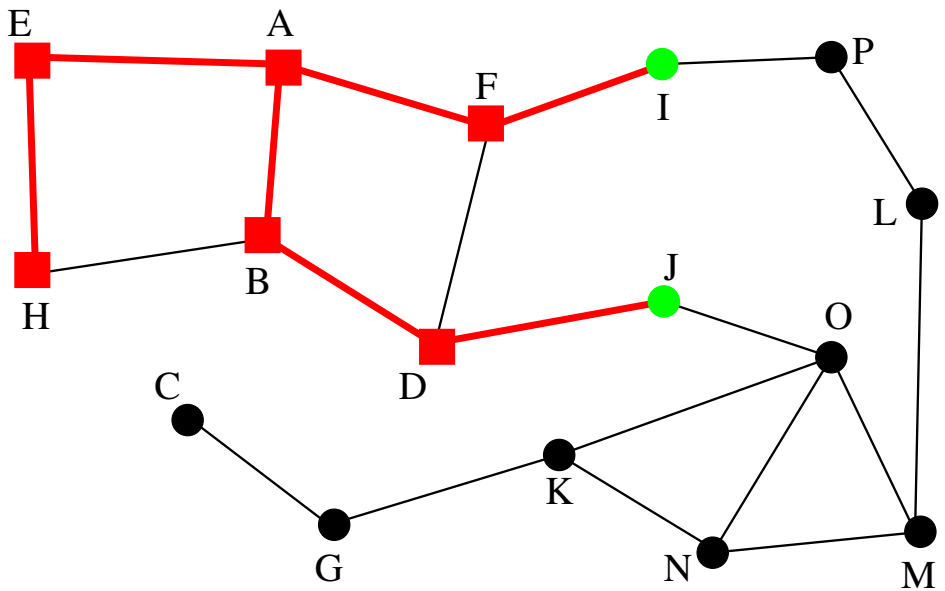
▸ Fin Animation

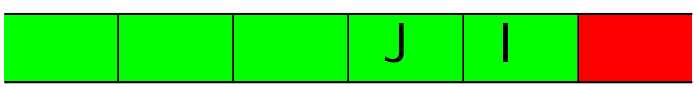




Exemple (parcours en largeur)

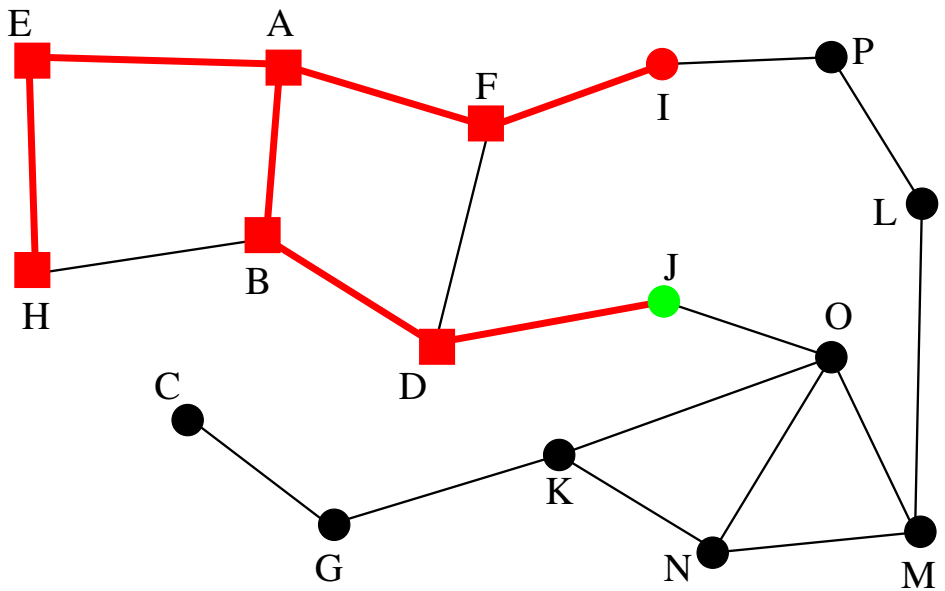
▸ Fin Animation





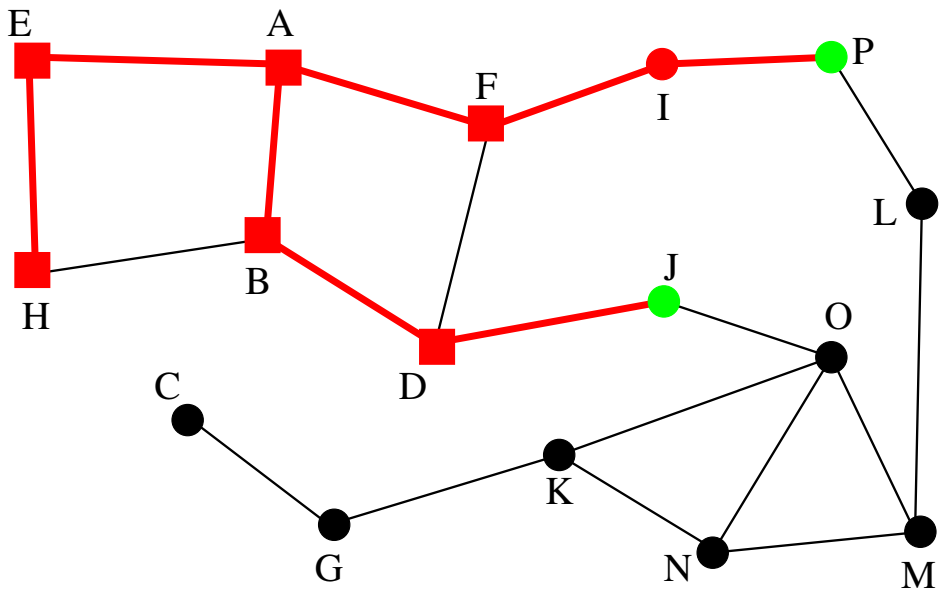
Exemple (parcours en largeur)

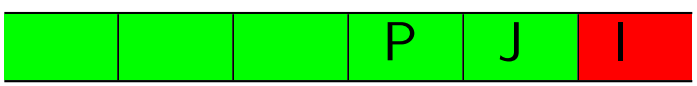
▸ Fin Animation



Exemple (parcours en largeur)

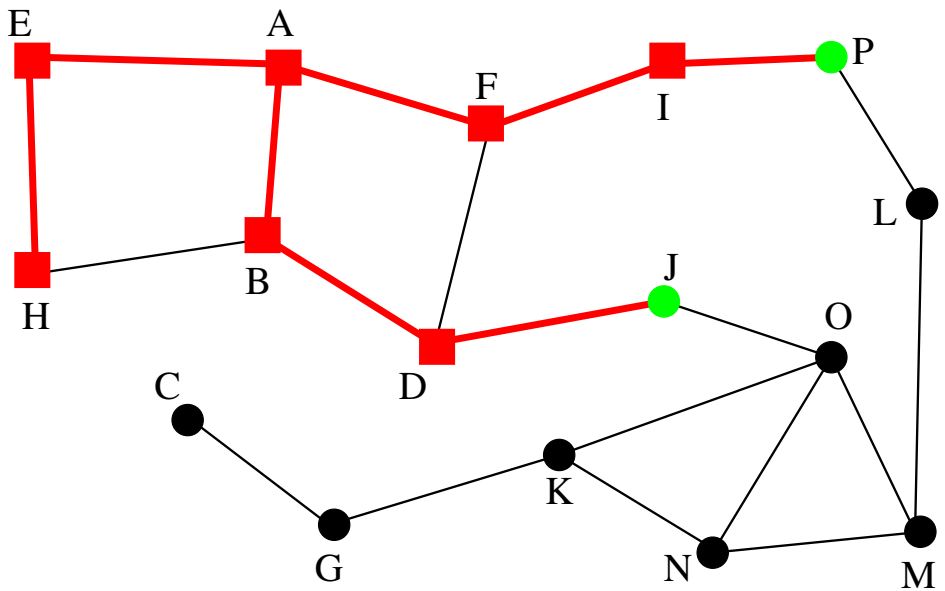
▸ Fin Animation

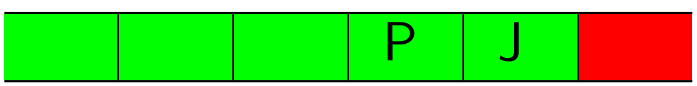




Exemple (parcours en largeur)

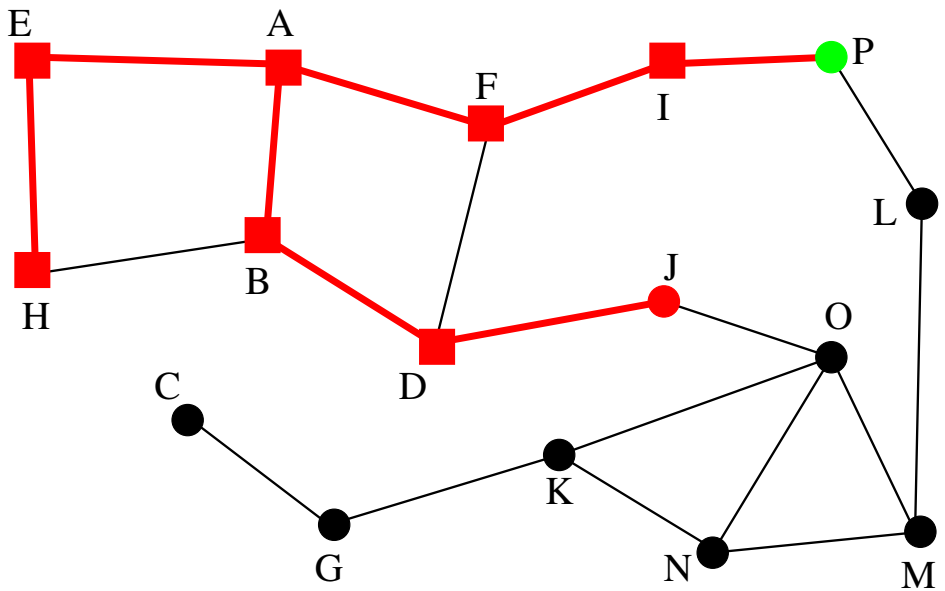
▸ Fin Animation





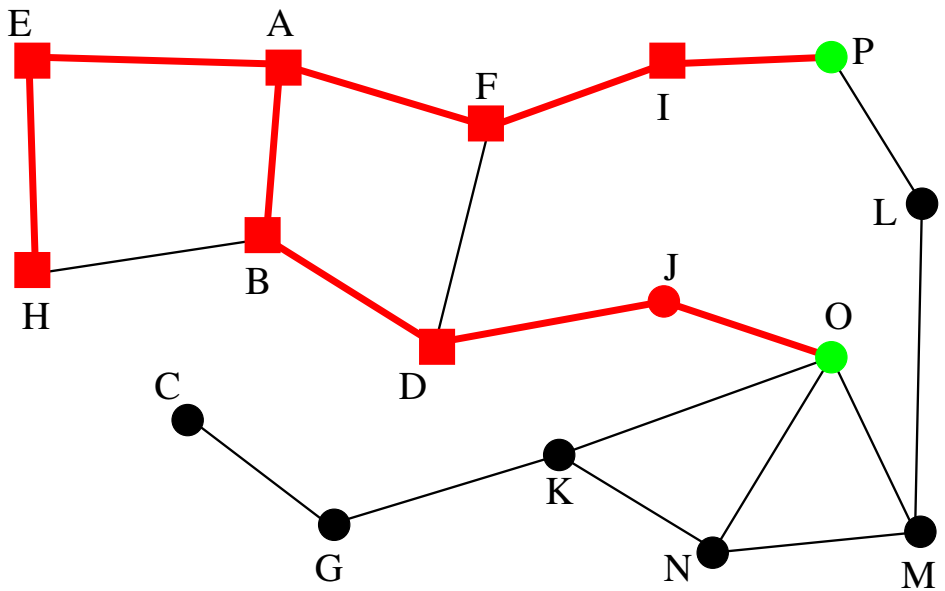
Exemple (parcours en largeur)

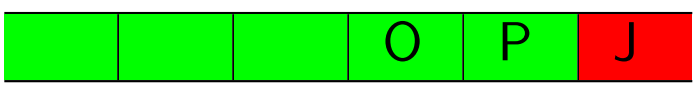
▸ Fin Animation



Exemple (parcours en largeur)

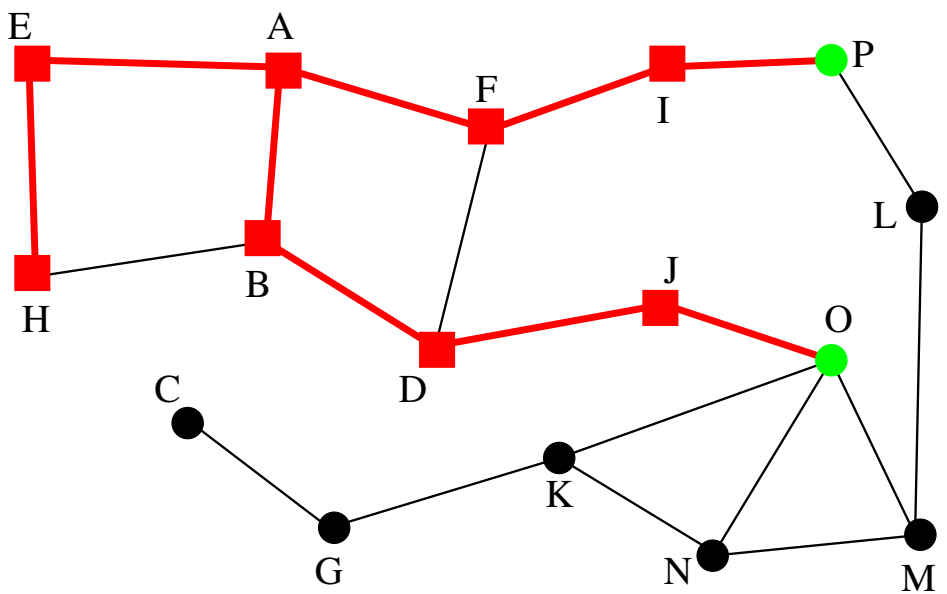
▸ Fin Animation

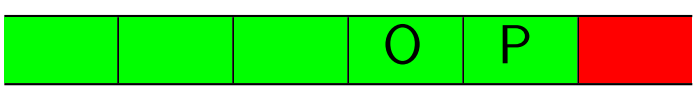




Exemple (parcours en largeur)

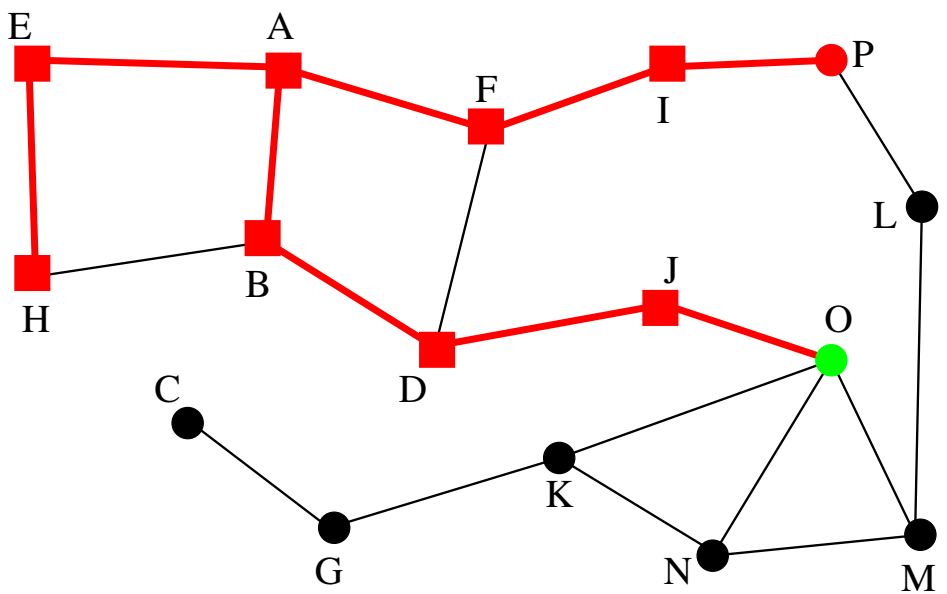
▸ Fin Animation





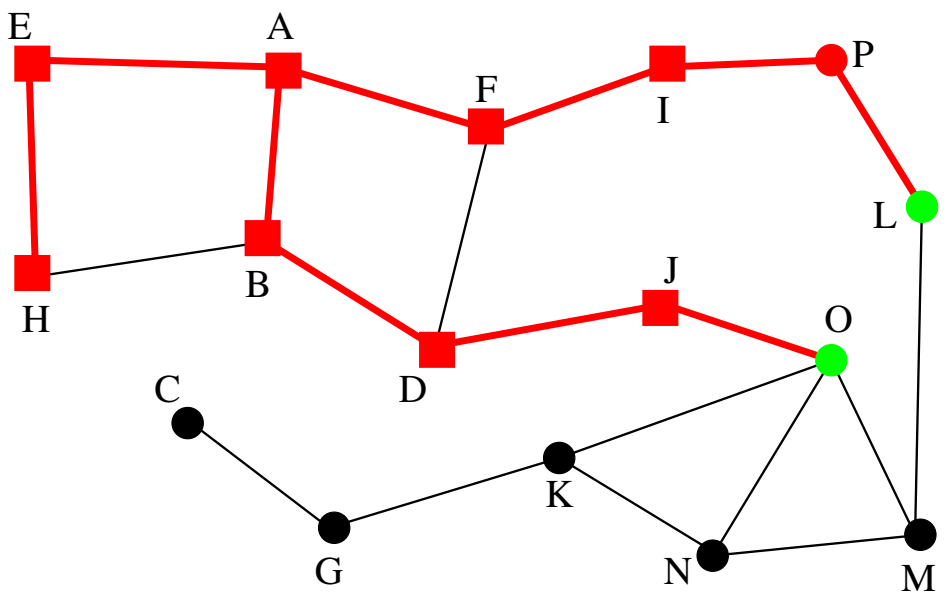
Exemple (parcours en largeur)

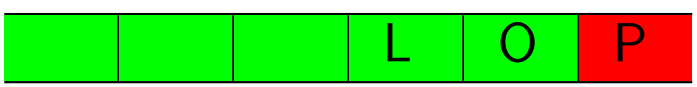
▸ Fin Animation



Exemple (parcours en largeur)

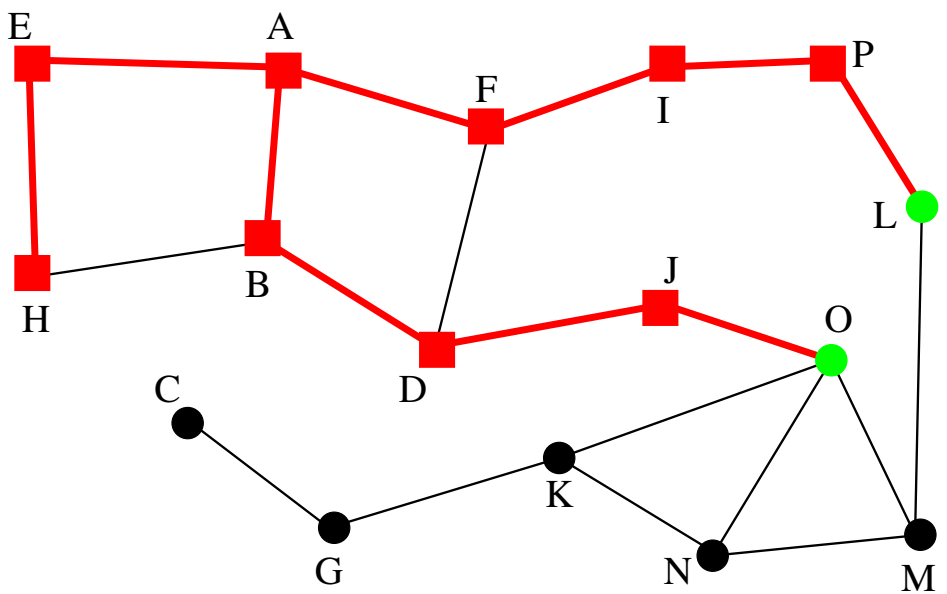
▸ Fin Animation





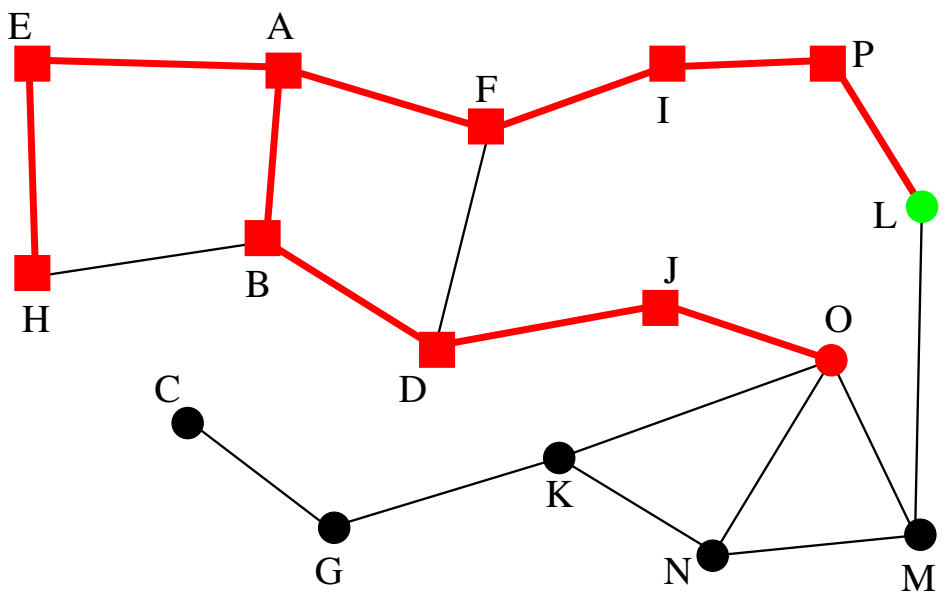
Exemple (parcours en largeur)

▸ Fin Animation



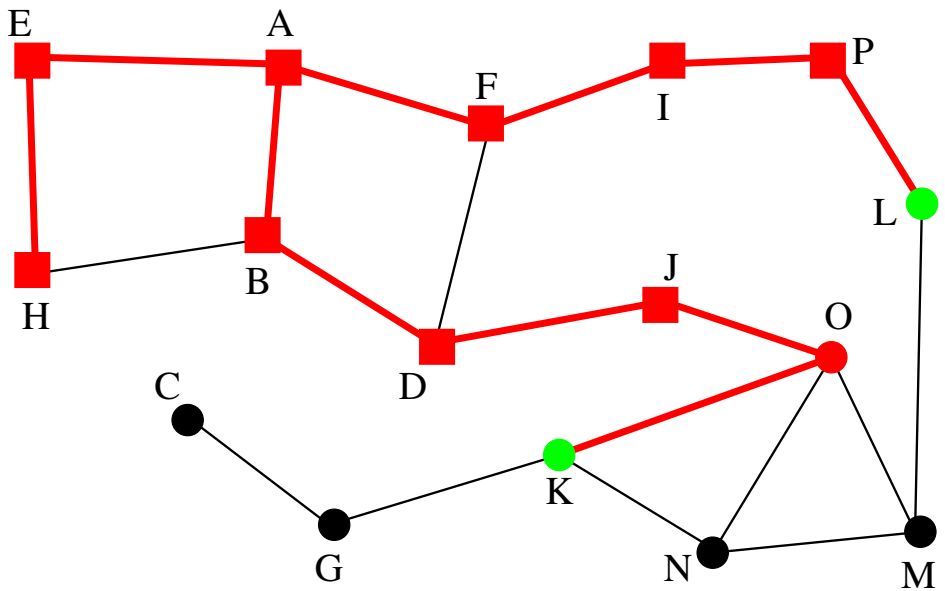
Exemple (parcours en largeur)

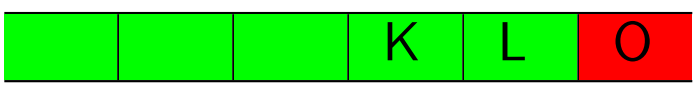
▸ Fin Animation



Exemple (parcours en largeur)

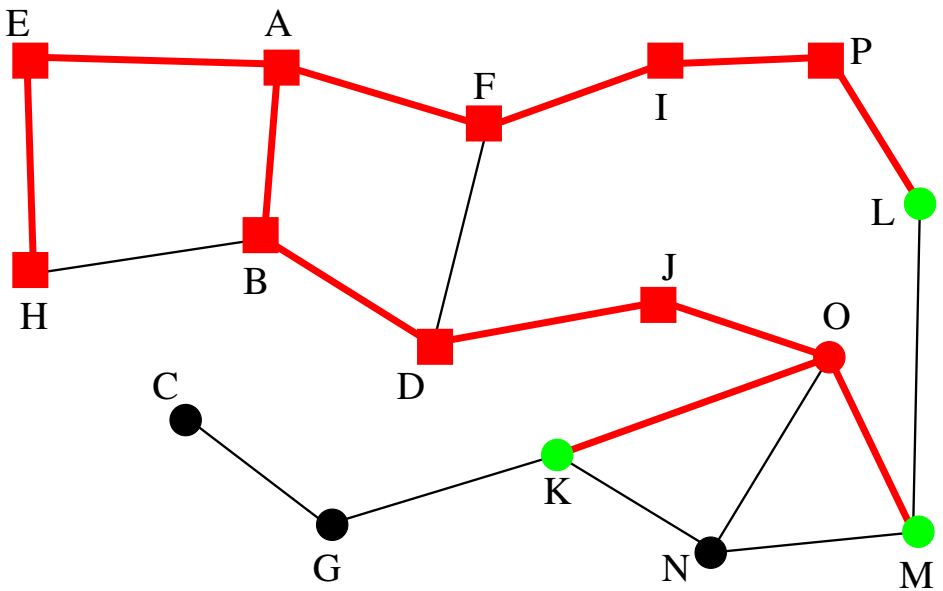
▸ Fin Animation





Exemple (parcours en largeur)

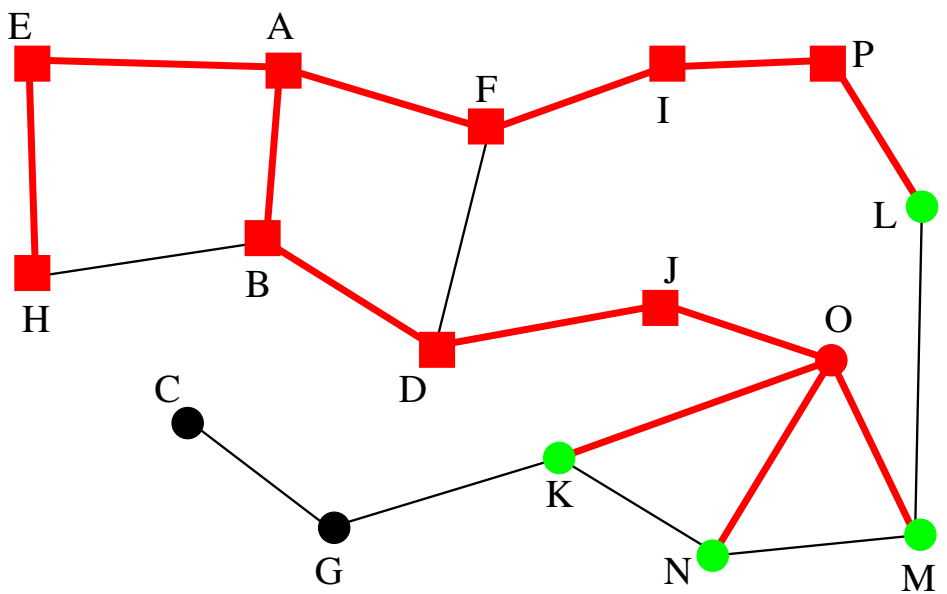
▸ Fin Animation



		M	K	L	O

Exemple (parcours en largeur)

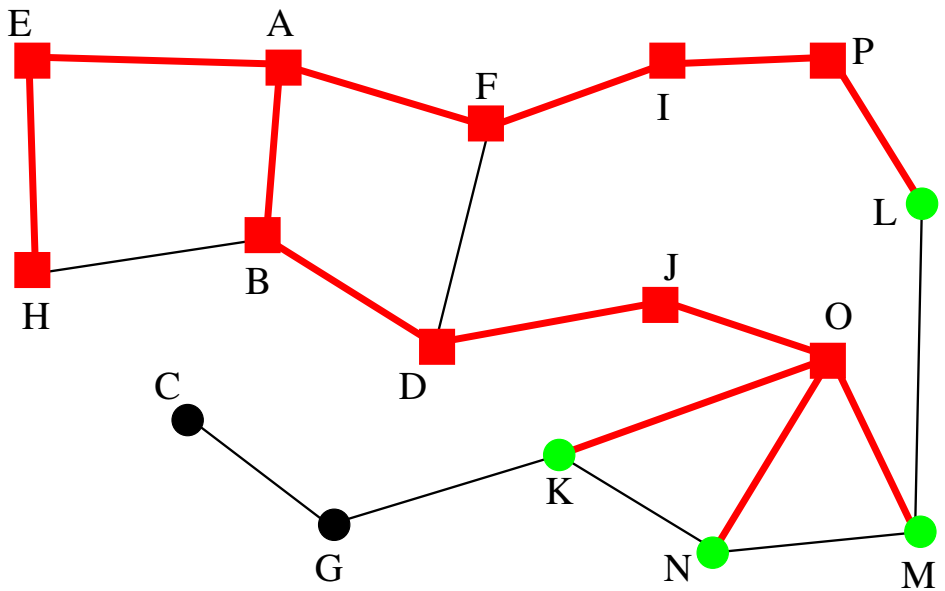
▸ Fin Animation

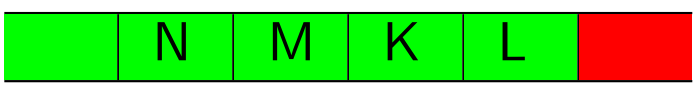


	N	M	K	L	O

Exemple (parcours en largeur)

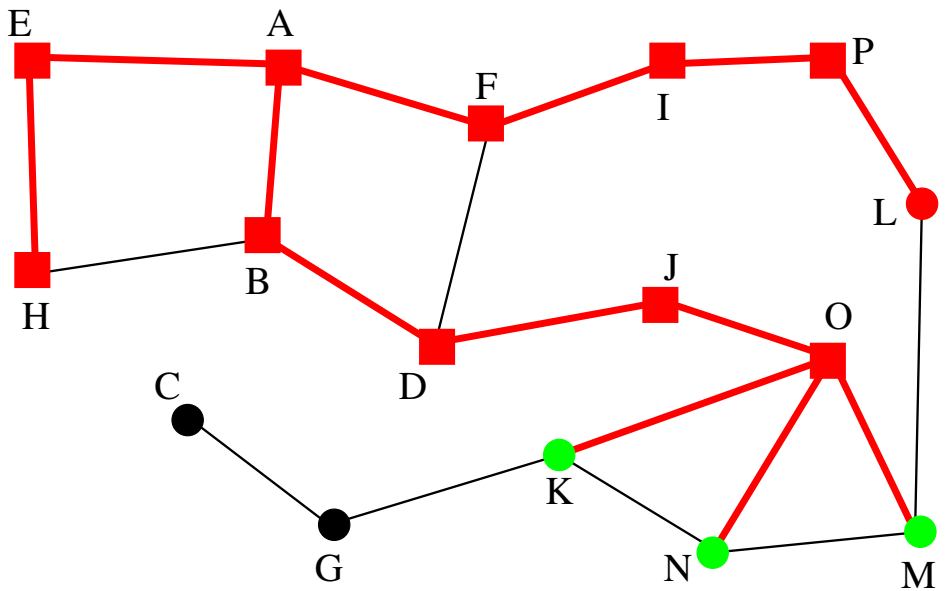
▸ Fin Animation





Exemple (parcours en largeur)

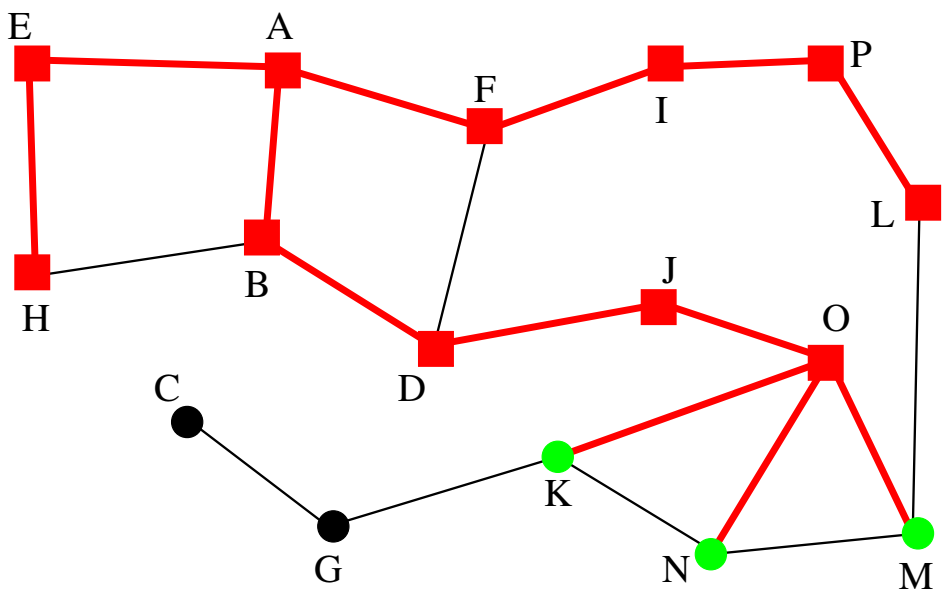
▸ Fin Animation

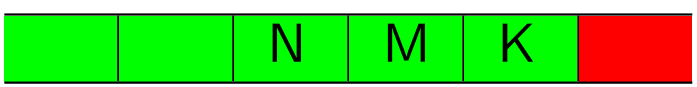


		N	M	K	L

Exemple (parcours en largeur)

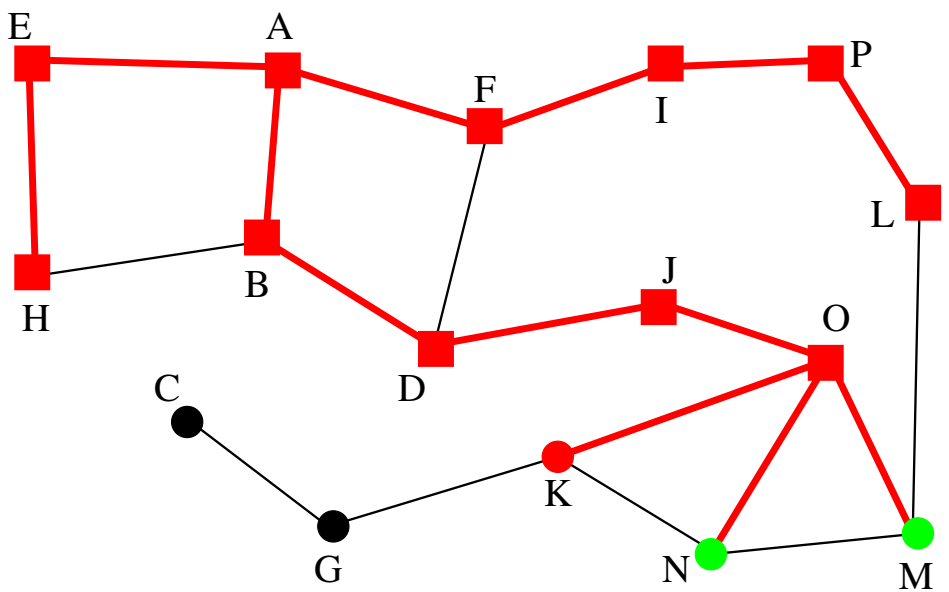
▸ Fin Animation

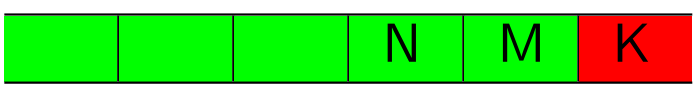




Exemple (parcours en largeur)

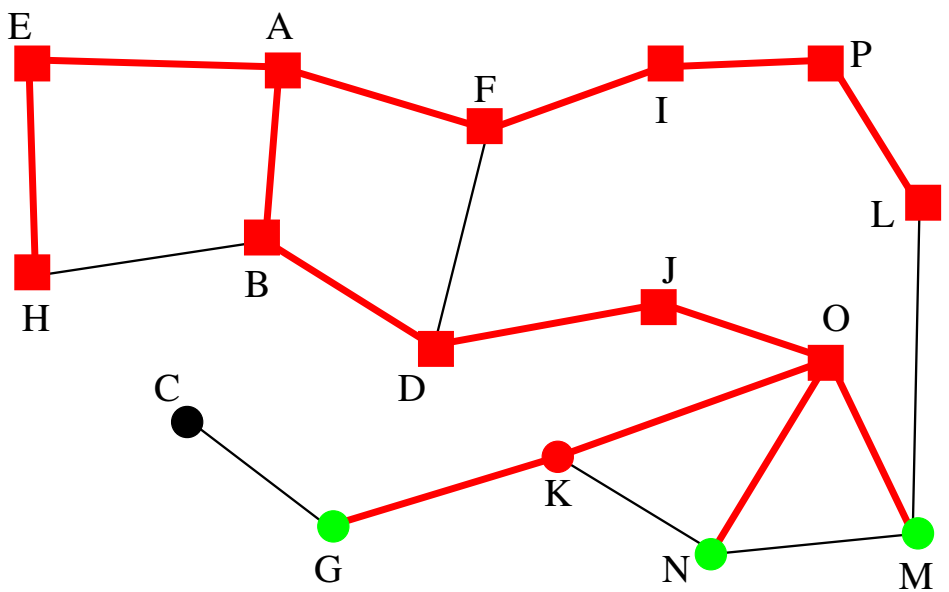
▸ Fin Animation





Exemple (parcours en largeur)

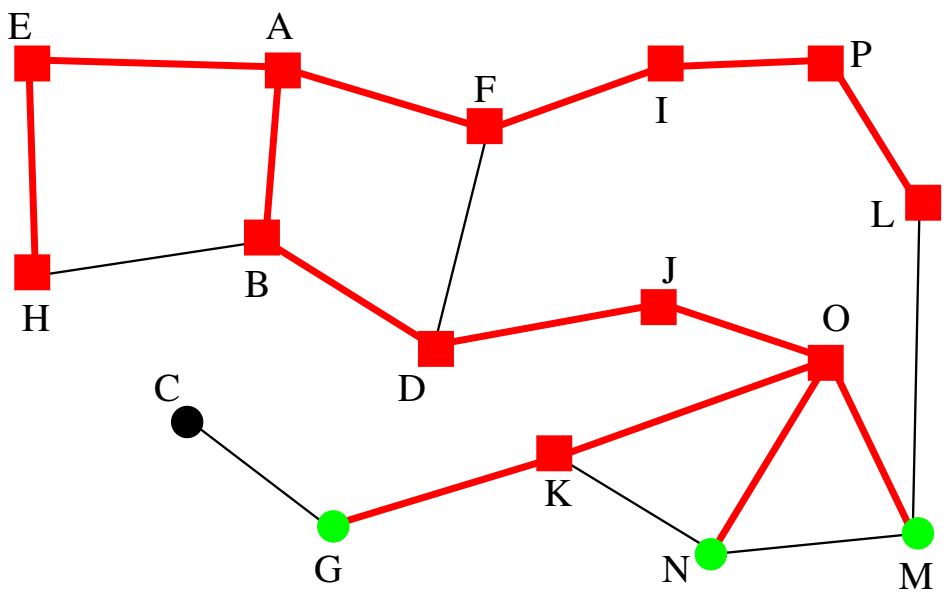
▸ Fin Animation



		G	N	M	K

Exemple (parcours en largeur)

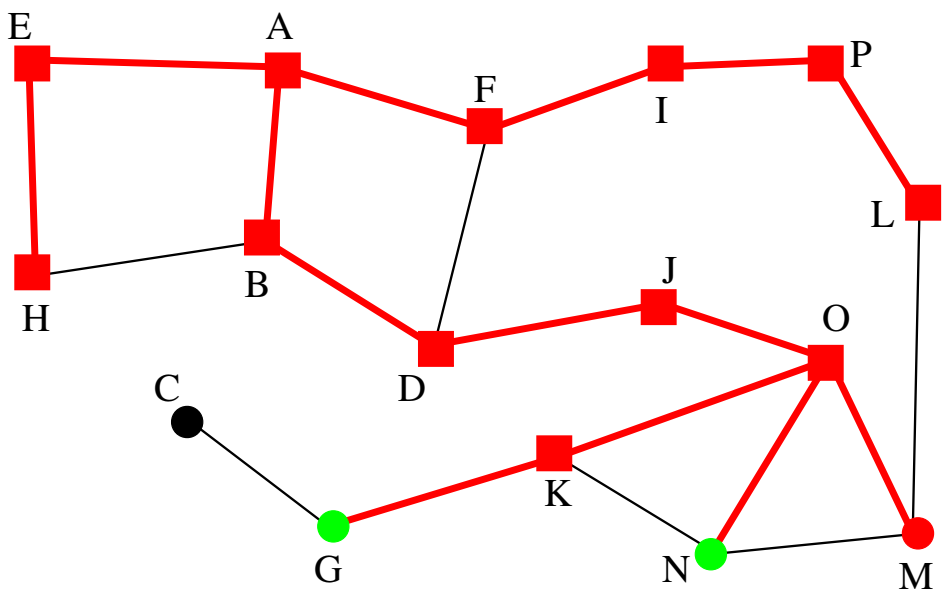
▸ Fin Animation



		G	N	M	

Exemple (parcours en largeur)

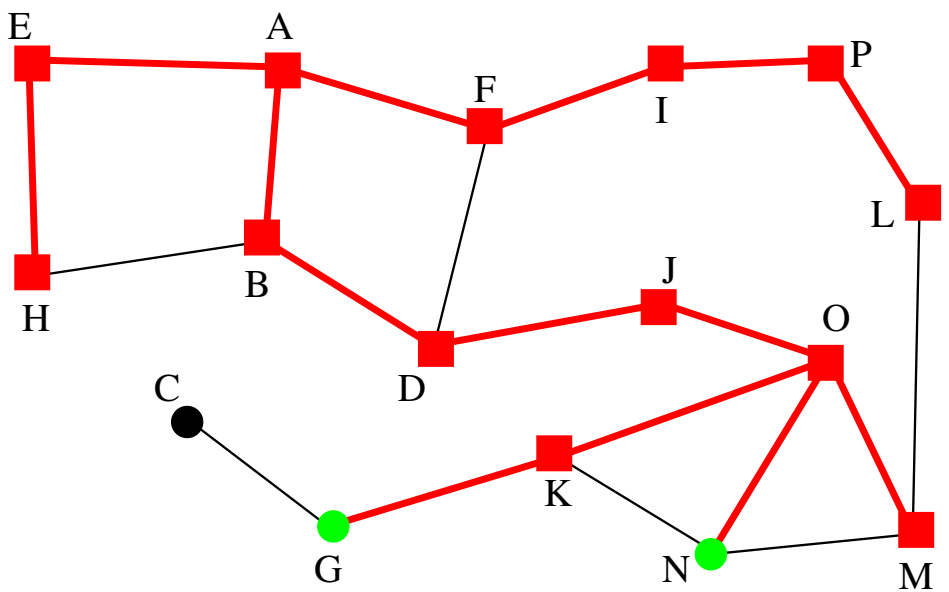
▸ Fin Animation

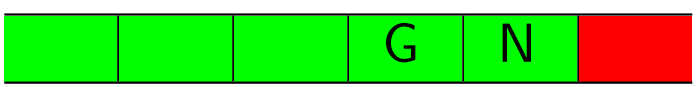


			G	N	M

Exemple (parcours en largeur)

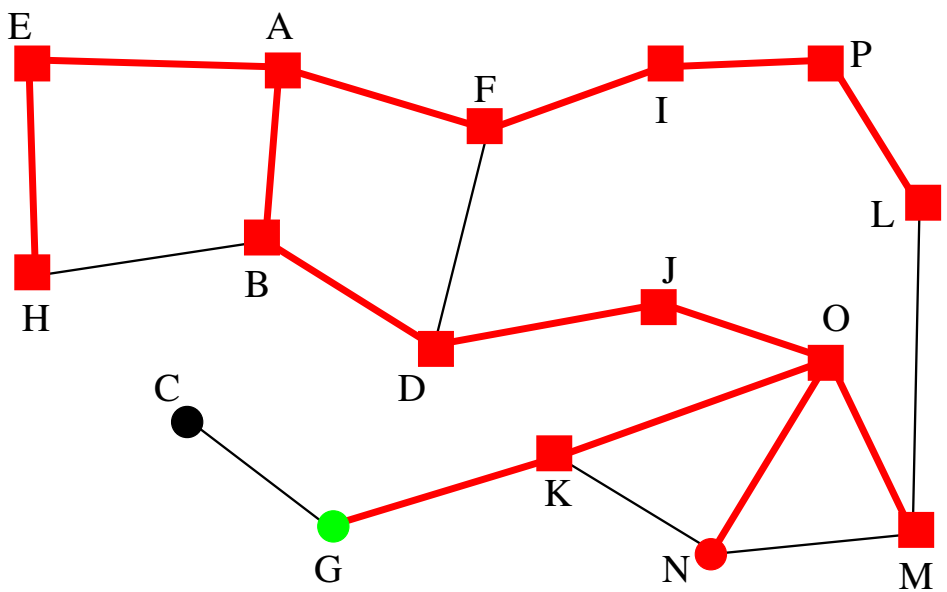
▸ Fin Animation





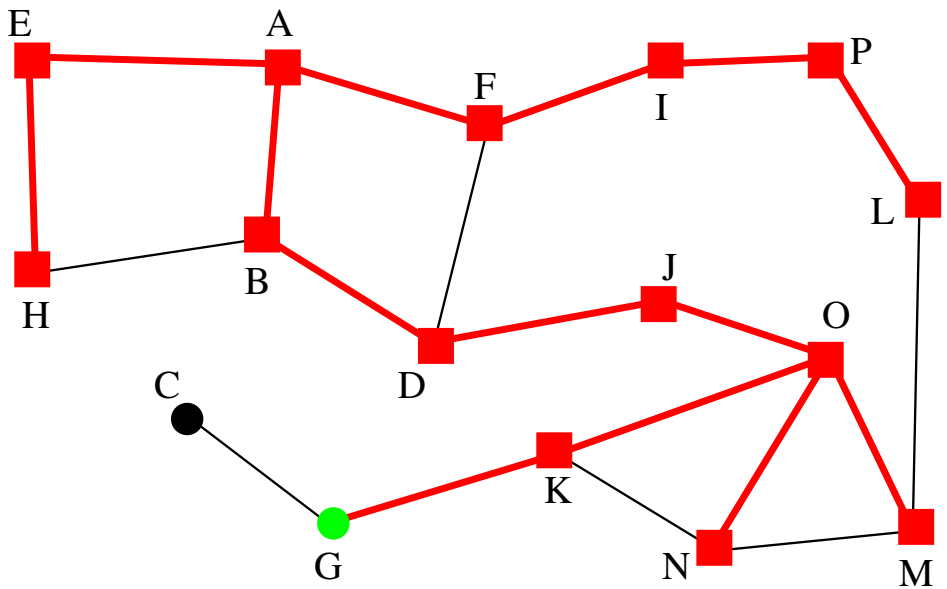
Exemple (parcours en largeur)

▸ Fin Animation



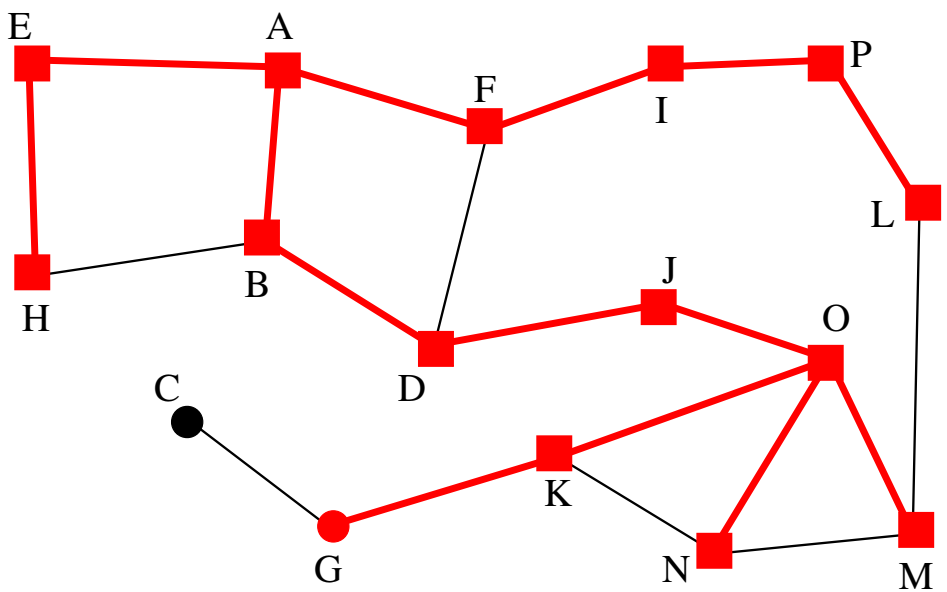
Exemple (parcours en largeur)

▸ Fin Animation



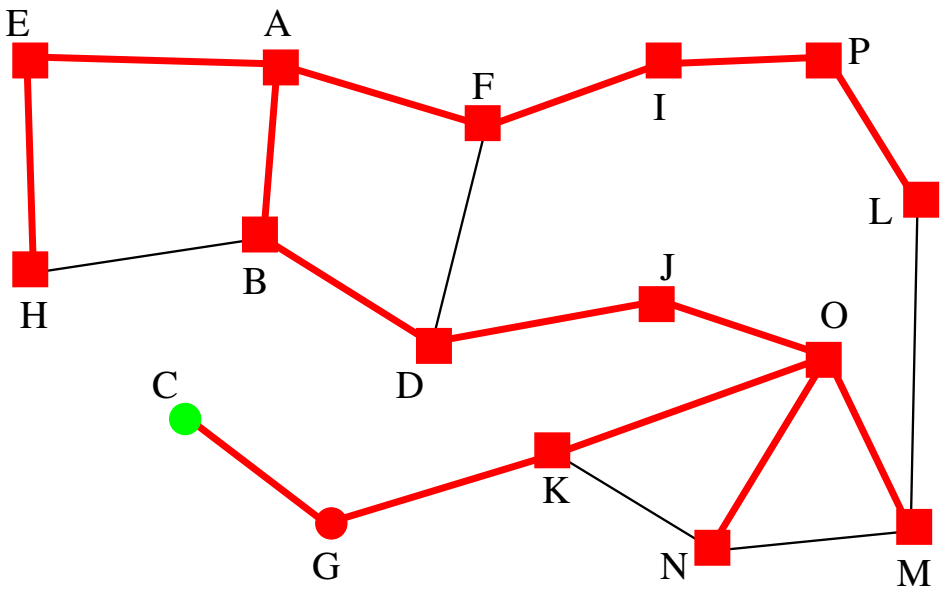
Exemple (parcours en largeur)

▸ Fin Animation



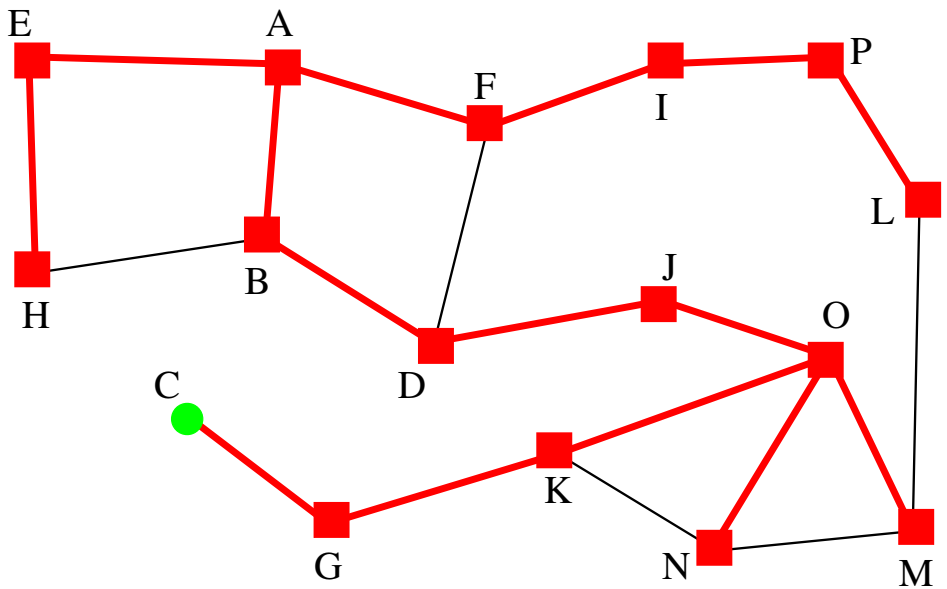
Exemple (parcours en largeur)

▸ Fin Animation



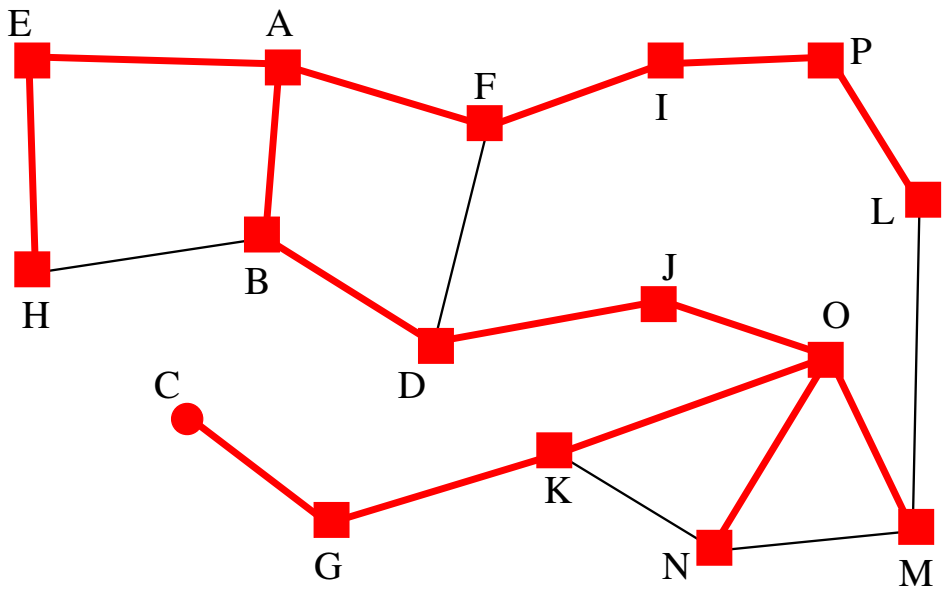
Exemple (parcours en largeur)

▸ Fin Animation



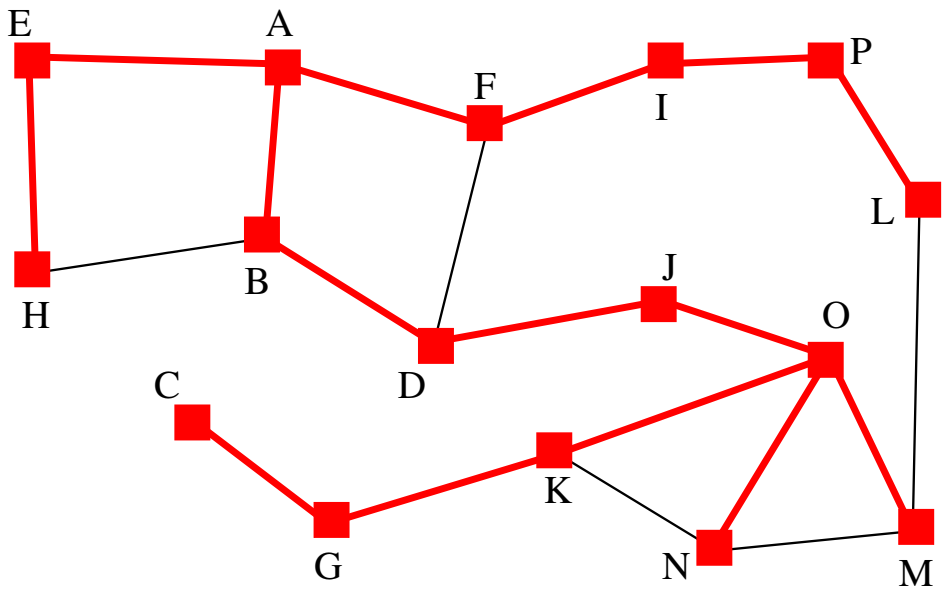
Exemple (parcours en largeur)

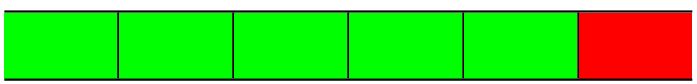
▸ Fin Animation



Exemple (parcours en largeur)

▸ Fin Animation





Algorithme du parcours en largeur

Entrées

- G non orienté
- s source

Variables locales

- F file (frontière)
- S ensemble des sommets connus
- u, v des sommets adjacents

Initialisation

- Ajouter s devant la file
- Ajouter s à S

Tant que la file n'est pas vide, répéter

- prendre le premier sommet v de la file d'attente,
- ajouter chaque voisin inconnu u de v à l'arrière de la file et ajouter u à S
- enlever v de la file

Algorithme 1 : Parcours en largeur

Données *G un graphe s un sommet de G*
Variables locales
S un ensemble `/* zone connue */`
F une file `/* la frontière */`
u, v deux sommets
début
  initialisation
  `ajouter(`*s*`,`*S*`)`
  `enfiler(`*s*`,`*F*`)`
  répéter
    v :=`valeurTete(`*F*`)`
    si *il existe u* $\notin$ *S adjacent à v* **alors**
      `ajouter(`*u*`,`*S*`)` `/* première visite de u */`
      `enfiler(`*u*`,`*F*`)`
    sinon
      `defiler(`*F*`)` `/* dernière visite de v */`
    fin si
  jusqu'à `estVide(`*F*`)`
fin

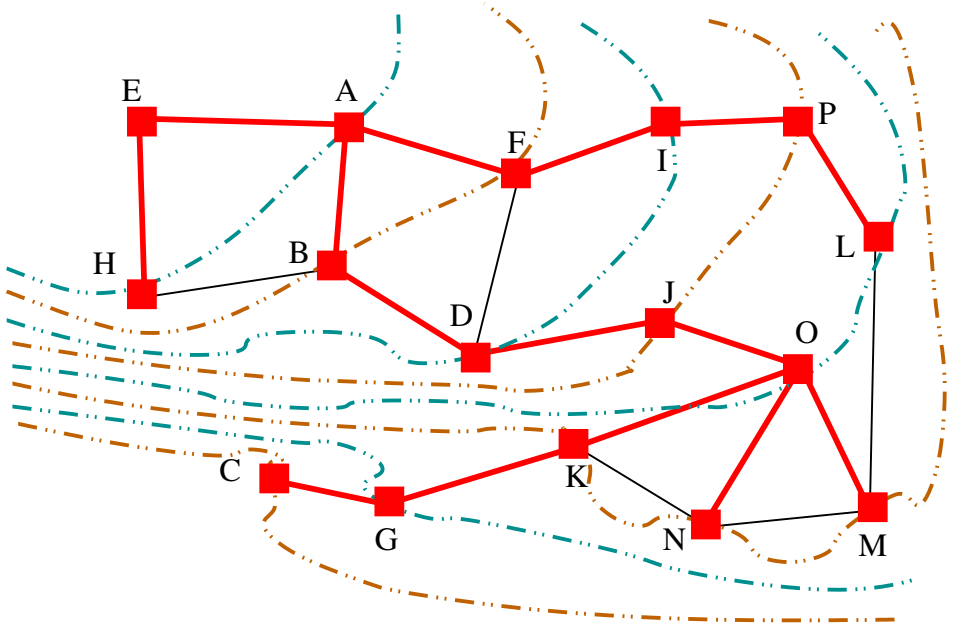
Sommaire

1. Introduction
2. Exemple
3. File et pile
4. Parcours en largeur
   - Rappel sur les files
   - Algorithme
   - Arbre de parcours et calcul des distances
5. Parcours en profondeur
   - Rappel sur les piles
   - Algorithme
   - Arbre de parcours
6. Pour finir

Propriétés d'un arbre de parcours en largeur :

Théorème 1 (Distance et Niveau)

Le niveau d'un sommet dans l'arbre est égal à la distance à la source dans le graphe. Autrement dit, les distances à la source dans l'arbre et dans le graphe sont les mêmes.



Preuve du théorème

- Si un sommet est au niveau d de l'arbre, alors il existe un chemin de longueur d de ce sommet à la source (par l'unique branche remontant à la source). Donc la distance est plus petite ou égale au niveau.
- On peut montrer l'inégalité inverse par récurrence sur le nombre de niveaux/les étapes de l'algorithme.
  - à la première étape, les voisins de la source sont ajoutés au niveau 1 de l'arbre.
  - supposons l'égalité vraie au rang n. Un sommet u à distance $n+1$ est voisin d'un sommet v à distance n. Or le sommet v est au niveau n de l'arbre par hypothèse et u ne peut pas avoir été ajouté avant v. L'algorithme du parcours en largeur va donc ajouter u au niveau $n+1$.
- Conclusion : on a bien distance=niveau

Algorithme 2 : Parcours en largeur + distance à la source

Variables locales

L tableau des distances à la source `/* indexés par les sommets */`

début

initialisation

`ajouter(`s`,`S`)`

`enfiler(`s`,`F`)`

$L[s] := 0$

répéter

$v :=$`valeurTete` (F)

si *il existe* $u \notin S$ *adjacent à* v **alors**

`ajouter(`u`,`S`)` `/* première visite de` u `*/`

`enfiler(`u`,`F`)`

$L[u] := L[v] + 1$

sinon

`defiler(`F`)` `/* dernière visite de` v `*/`

fin si

jusqu'à `estVide(`F`)`

fin

Sorties : L, tableau des distances à la source s

Sommaire

1. Introduction
2. Exemple
3. File et pile
4. Parcours en largeur
   - Rappel sur les files
   - Algorithme
   - Arbre de parcours et calcul des distances
5. Parcours en profondeur
   - Rappel sur les piles
   - Algorithme
   - Arbre de parcours
6. Pour finir

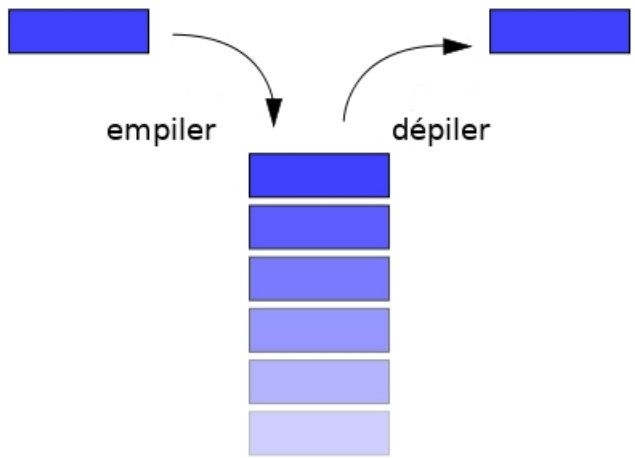
Sommaire

1. Introduction
2. Exemple
3. File et pile
4. Parcours en largeur
   - Rappel sur les files
   - Algorithme
   - Arbre de parcours et calcul des distances
5. Parcours en profondeur
   - Rappel sur les piles
   - Algorithme
   - Arbre de parcours
6. Pour finir

Définition d'une pile

Les primitives permettant de gérer une pile sont :

- Ajouter un sommet (toujours en haut de la pile) (`empiler`)
- Enlever un sommet (toujours en haut de la pile) (`dépiler`)
- Vérifier si la pile est vide (`estVide?`)



- On peut aussi juste regarder le sommet en haut de la pile sans l'enlever (`valeurSommetPile`)

Les piles en pratique

- En anglais : *Stack* ou bien LIFO *(Last In First Out)*
- Peut se coder facilement avec des listes chaînées (par exemple)
- Des librairies implémentant les piles existent dans la majeure partie des langages de programmation

Exemple

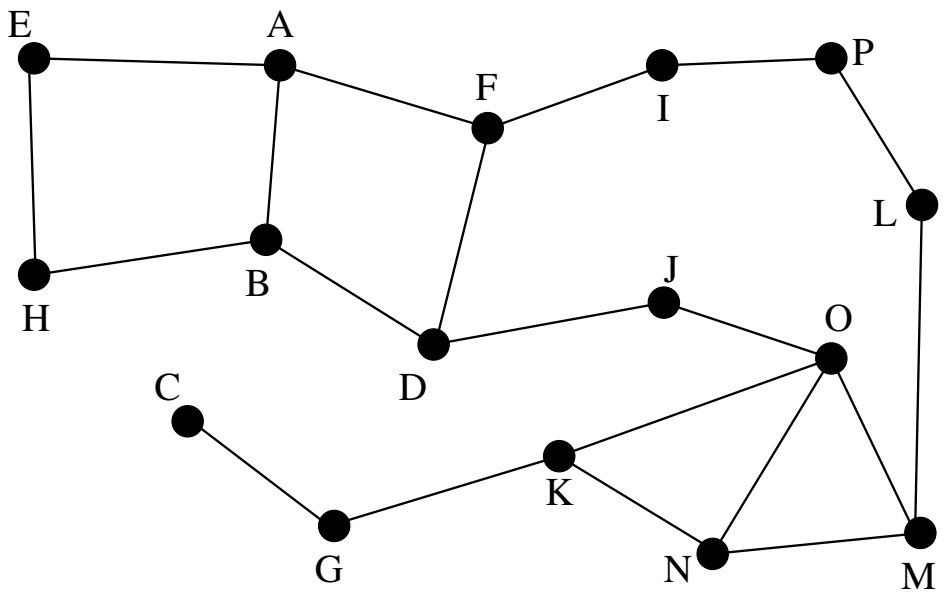
- On peut utiliser une pile si on veut implémenter *annuler une action*
- Une pile stockant les appels de fonctions est utilisée par la plupart des langages de programmation compilés

Sommaire

1. Introduction
2. Exemple
3. File et pile
4. Parcours en largeur
   - Rappel sur les files
   - Algorithme
   - Arbre de parcours et calcul des distances
5. Parcours en profondeur
   - Rappel sur les piles
   - Algorithme
   - Arbre de parcours
6. Pour finir

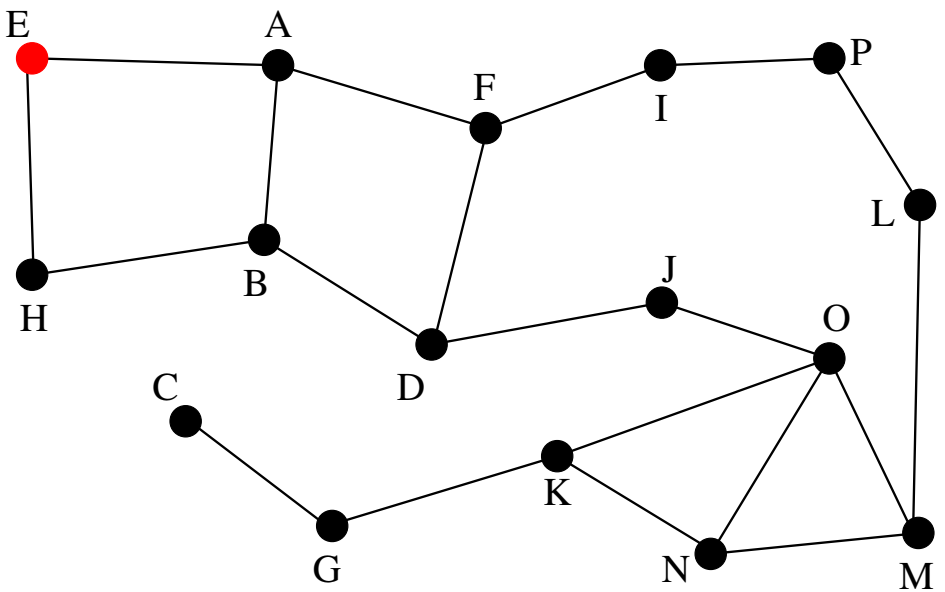
Exemple

Fin Animation



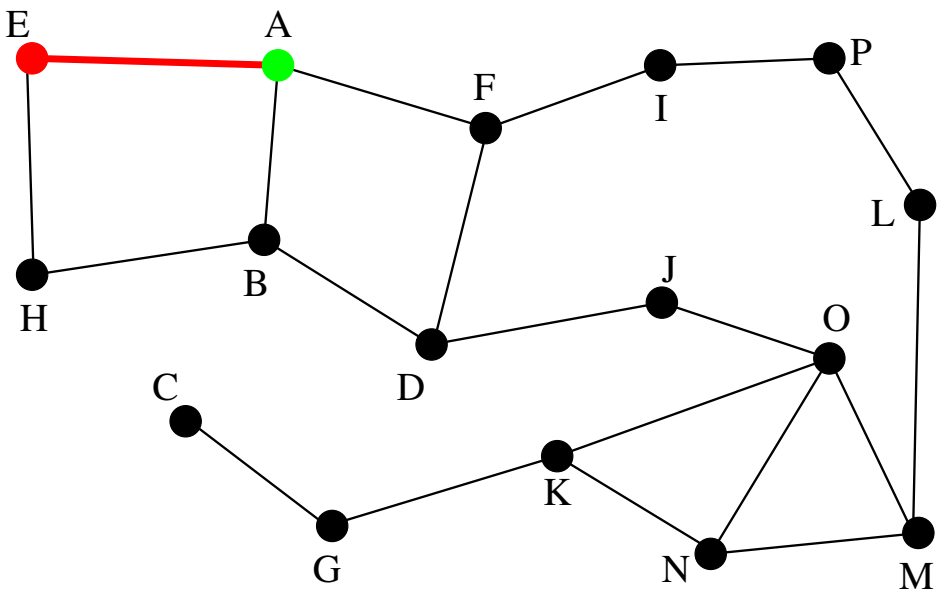
Exemple

Fin Animation



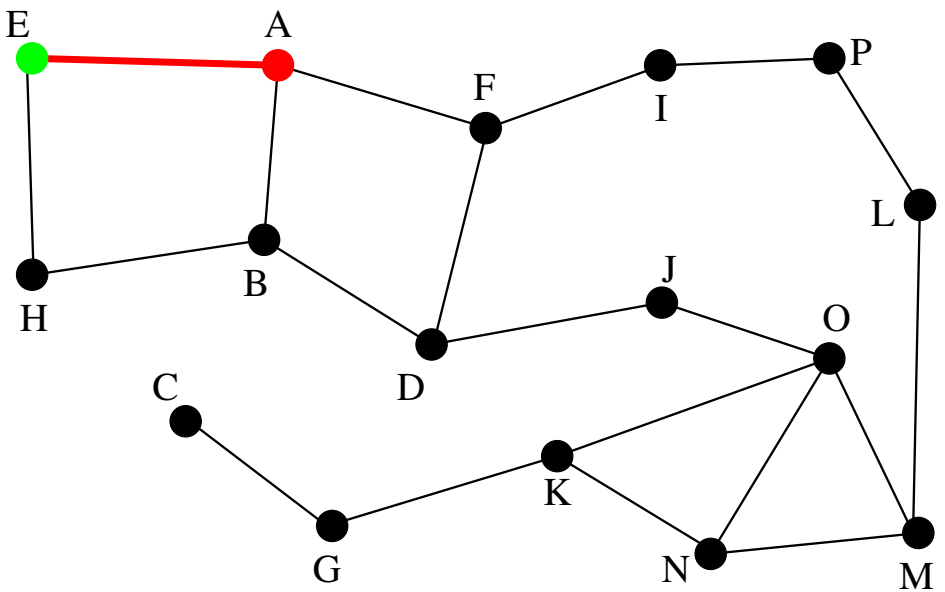
Exemple

▸ Fin Animation



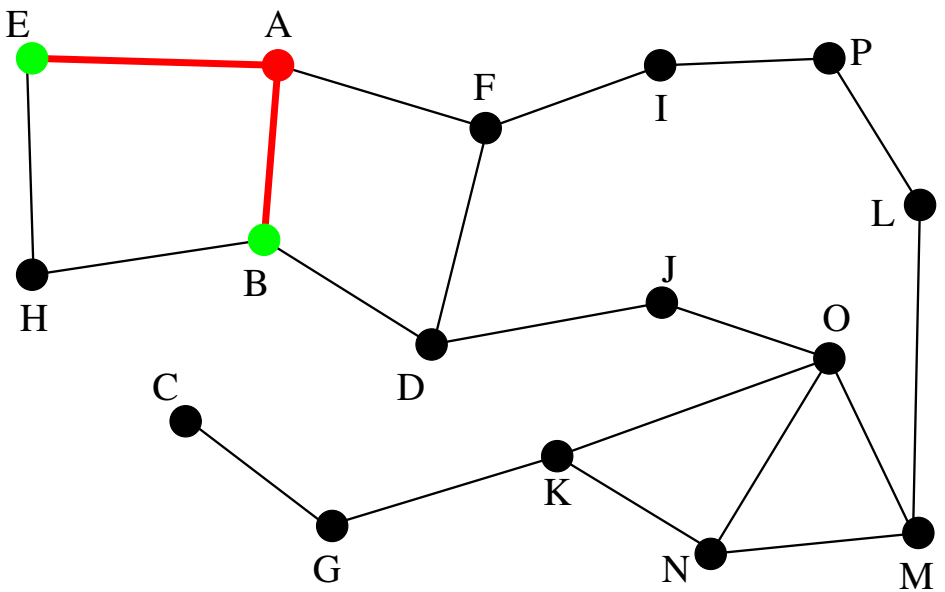
Exemple

Fin Animation



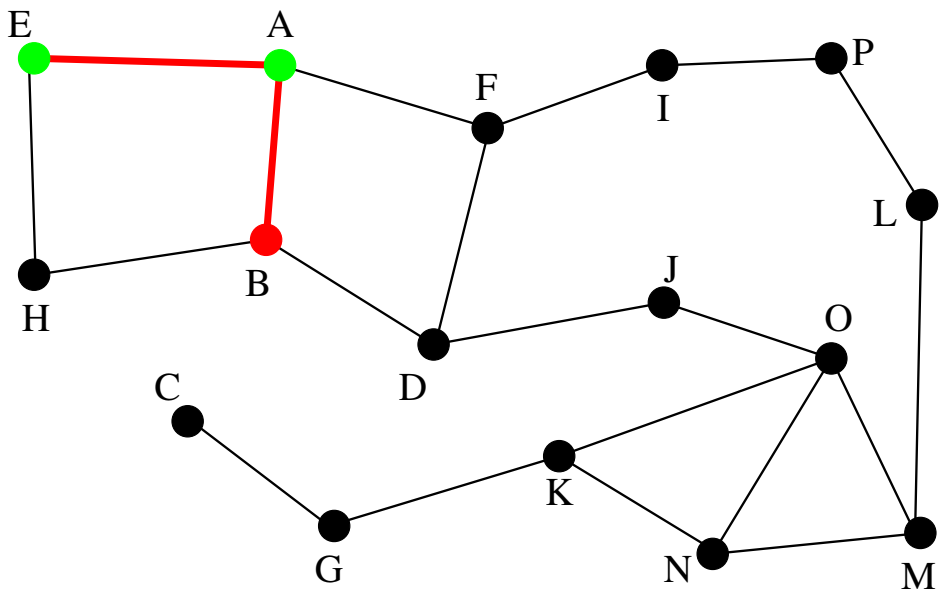
Exemple

Fin Animation



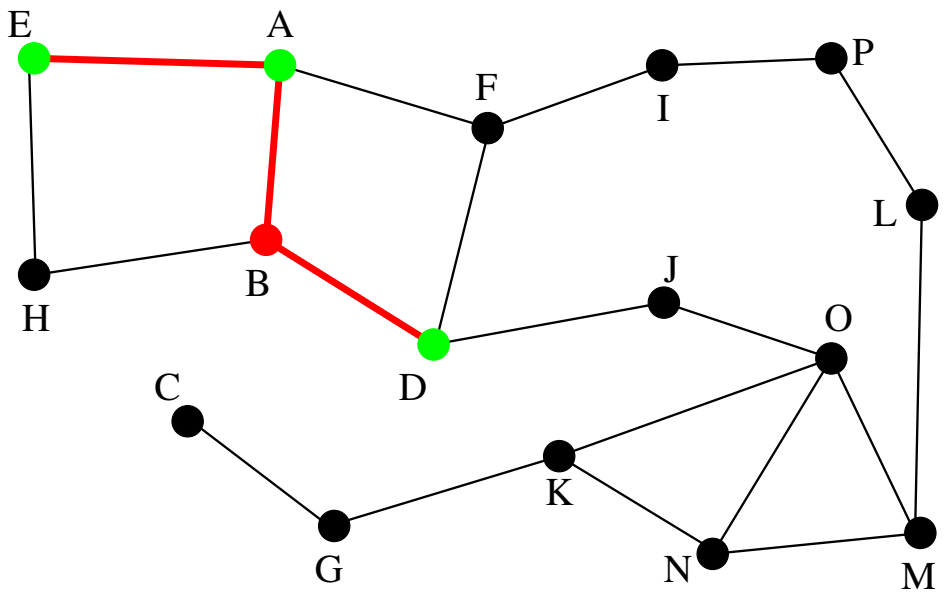
Exemple

▸ Fin Animation



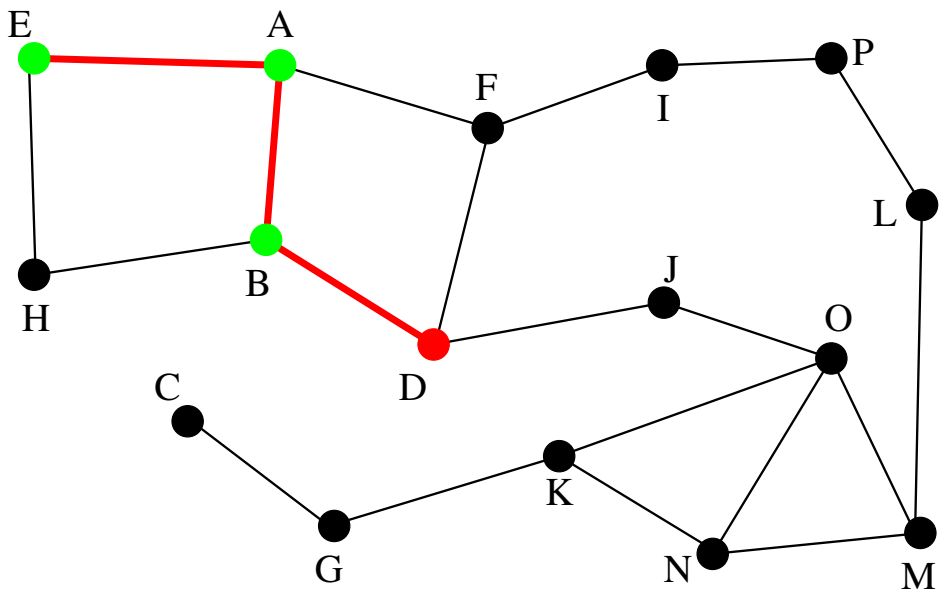
Exemple

▸ Fin Animation



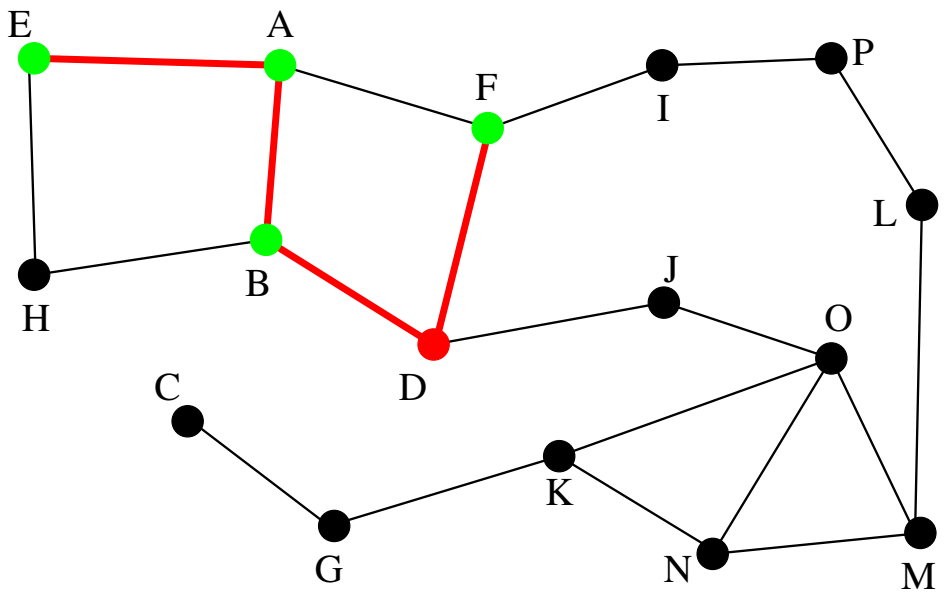
Exemple

▸ Fin Animation



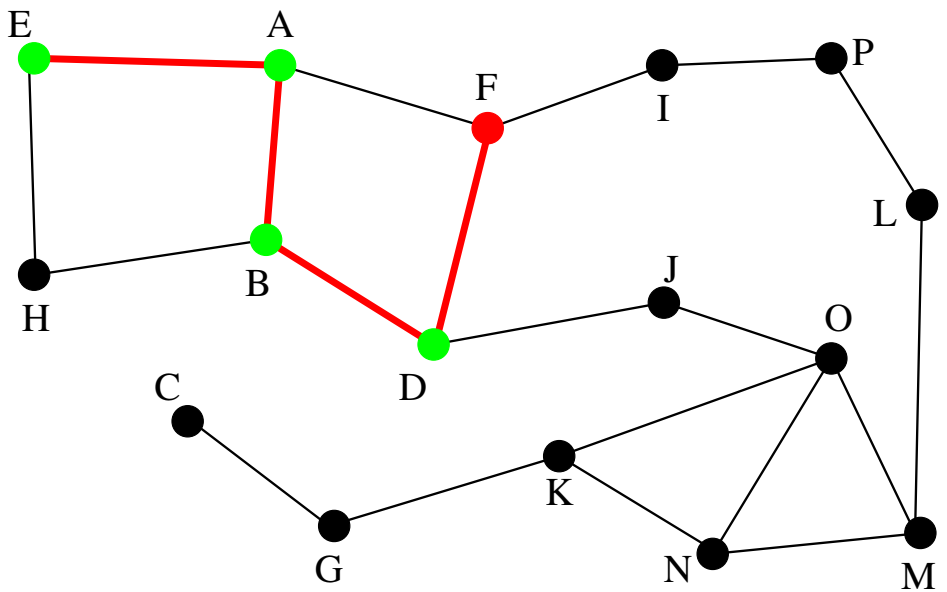
Exemple

▸ Fin Animation



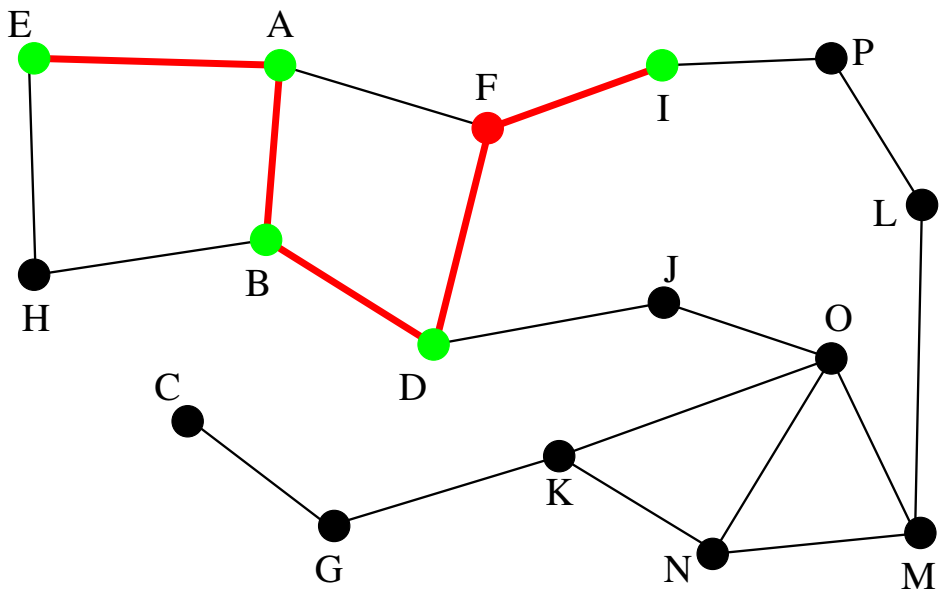
Exemple

Fin Animation



Exemple

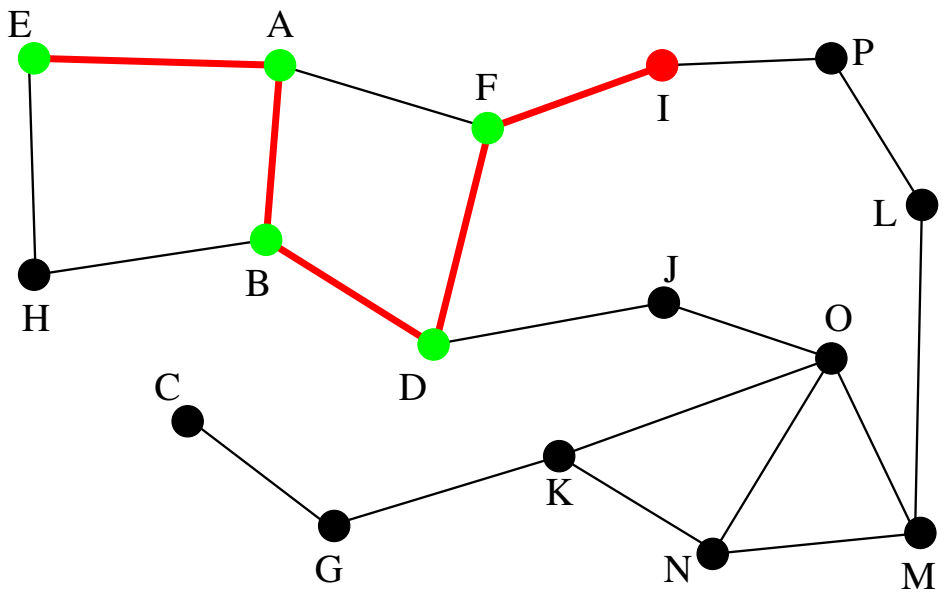
▸ Fin Animation



E	A	B	D	F	I									

Exemple

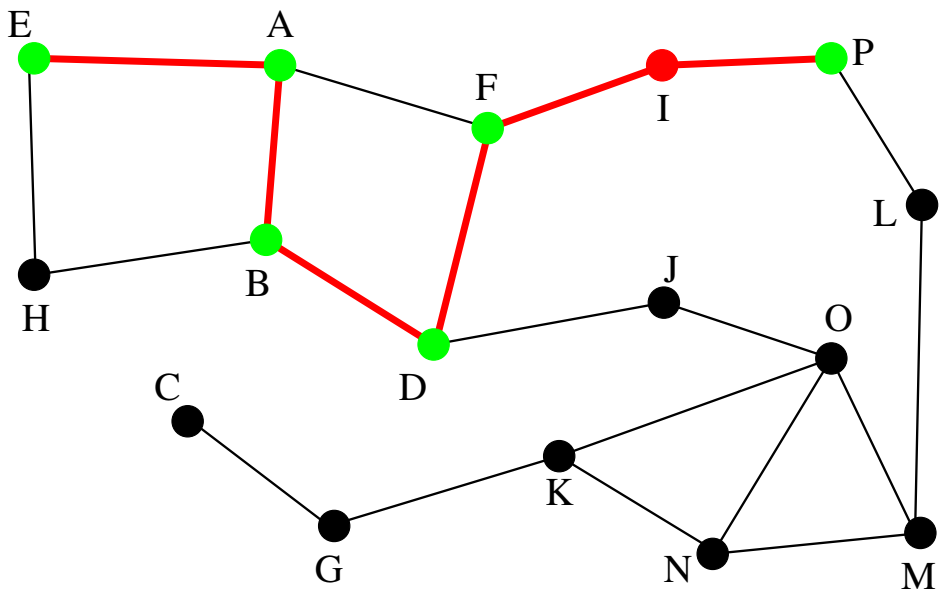
▸ Fin Animation



E	A	B	D	F	I									

Exemple

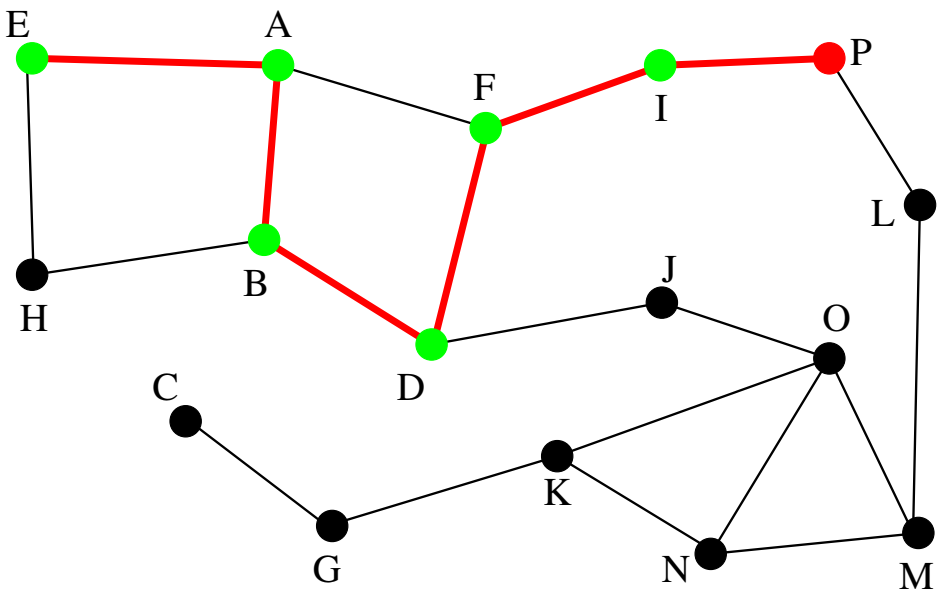
▸ Fin Animation



E	A	B	D	F	I	P								

Exemple

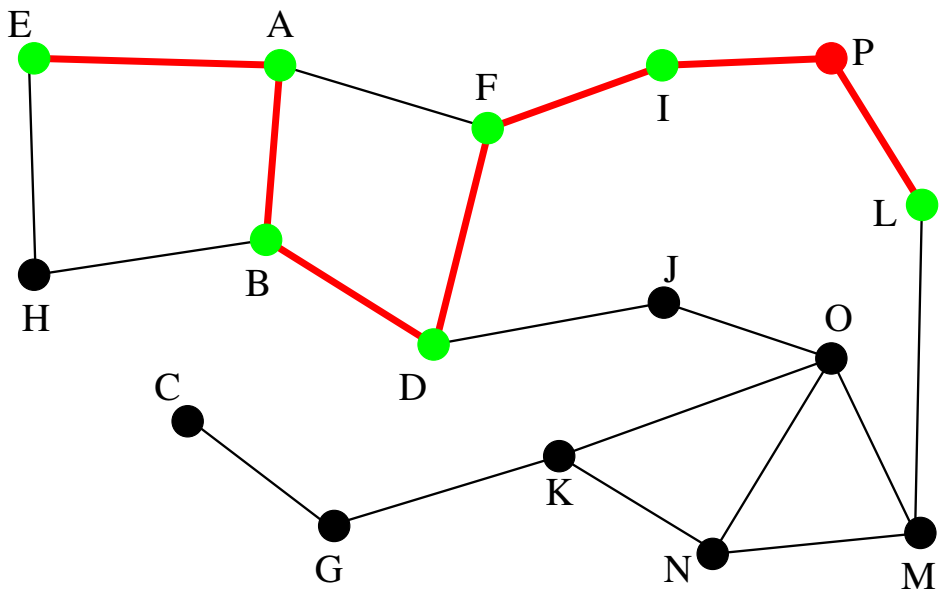
▸ Fin Animation



E	A	B	D	F	I	P								

Exemple

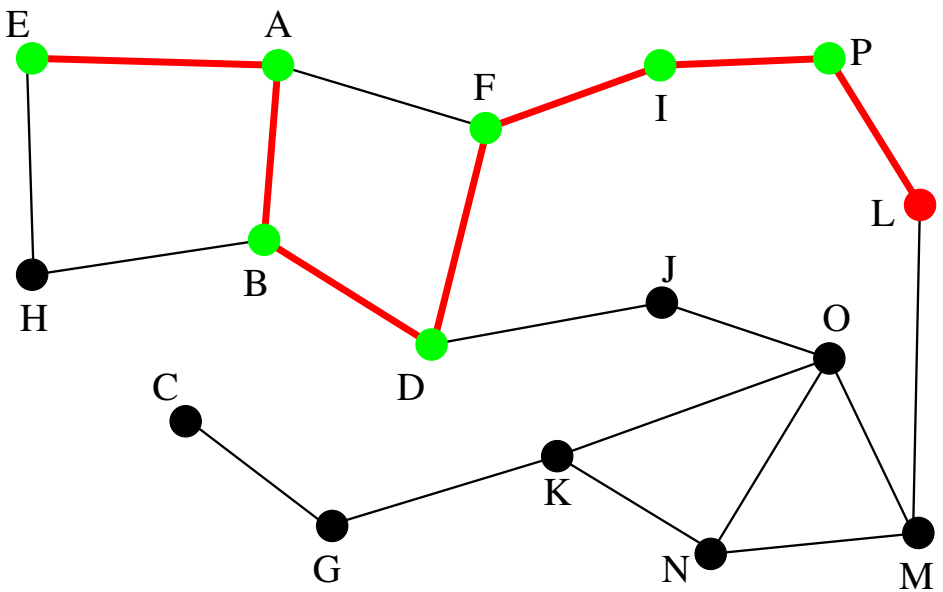
▸ Fin Animation



E	A	B	D	F	I	P	L							

Exemple

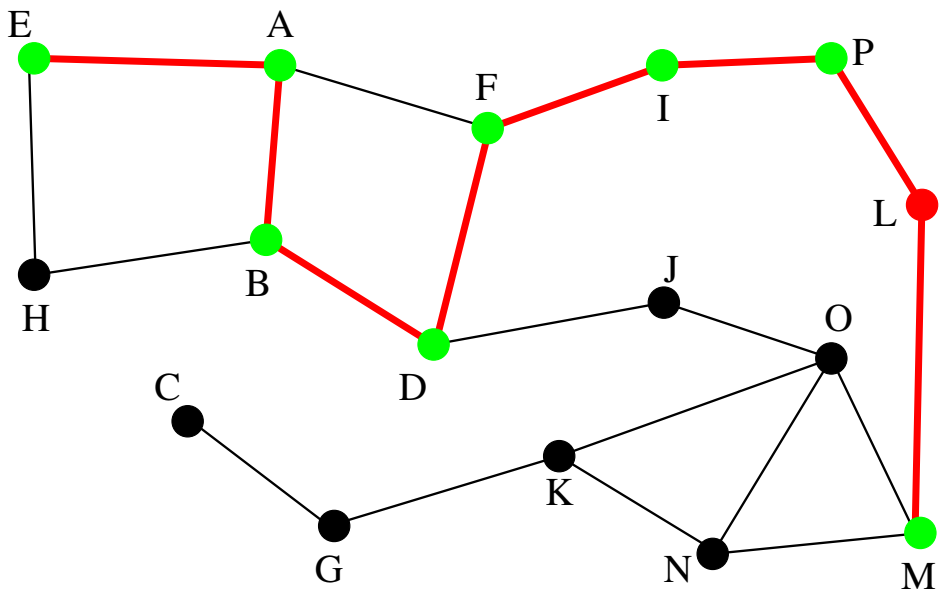
Fin Animation



E	A	B	D	F	I	P	L							

Exemple

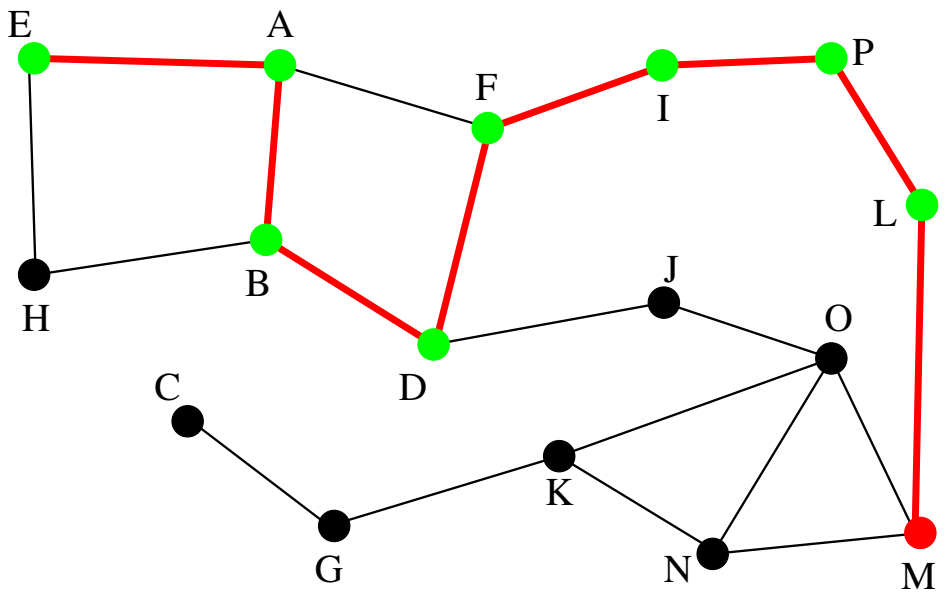
▸ Fin Animation



E	A	B	D	F	I	P	L	M						

Exemple

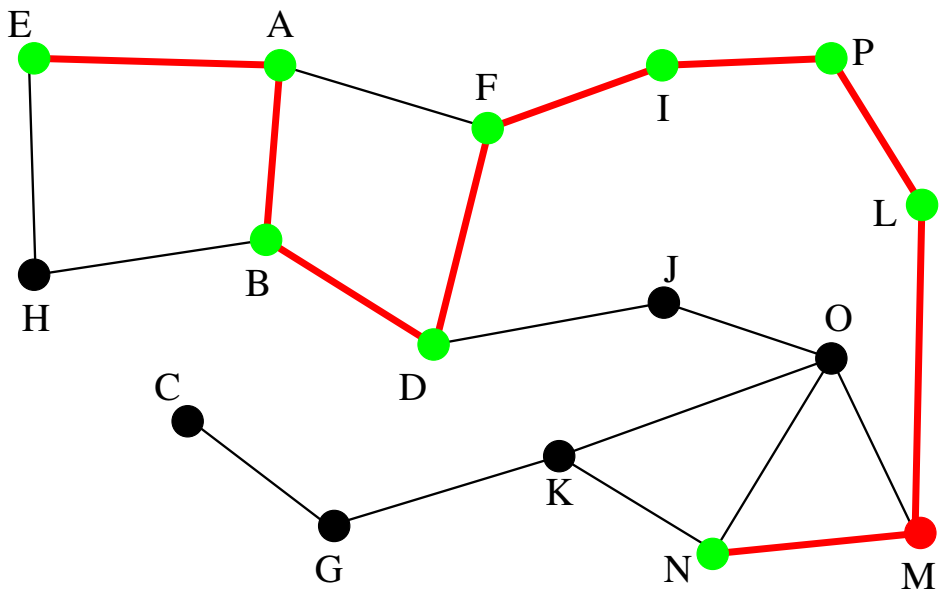
▸ Fin Animation



E	A	B	D	F	I	P	L	M						

Exemple

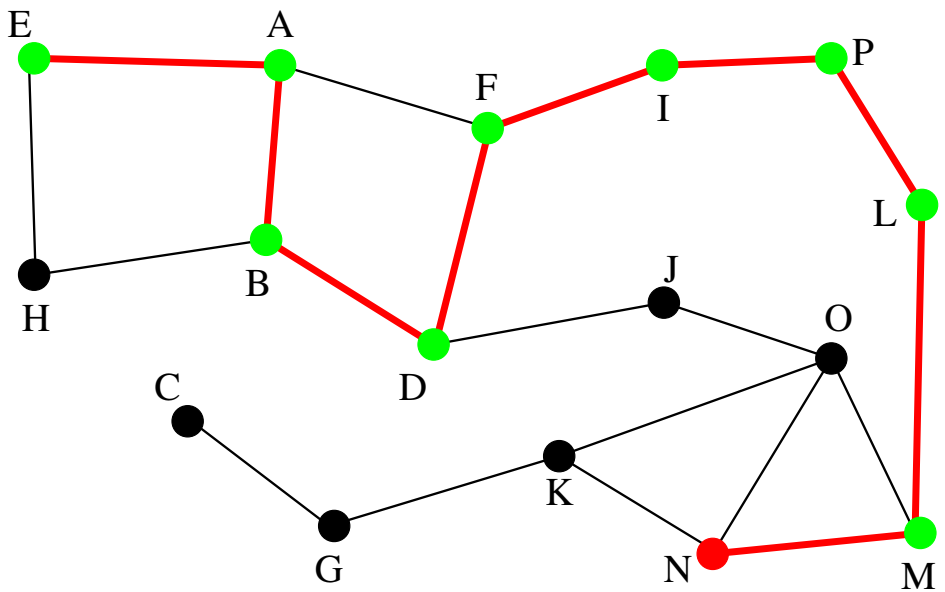
▸ Fin Animation



E	A	B	D	F	I	P	L	M	N					

Exemple

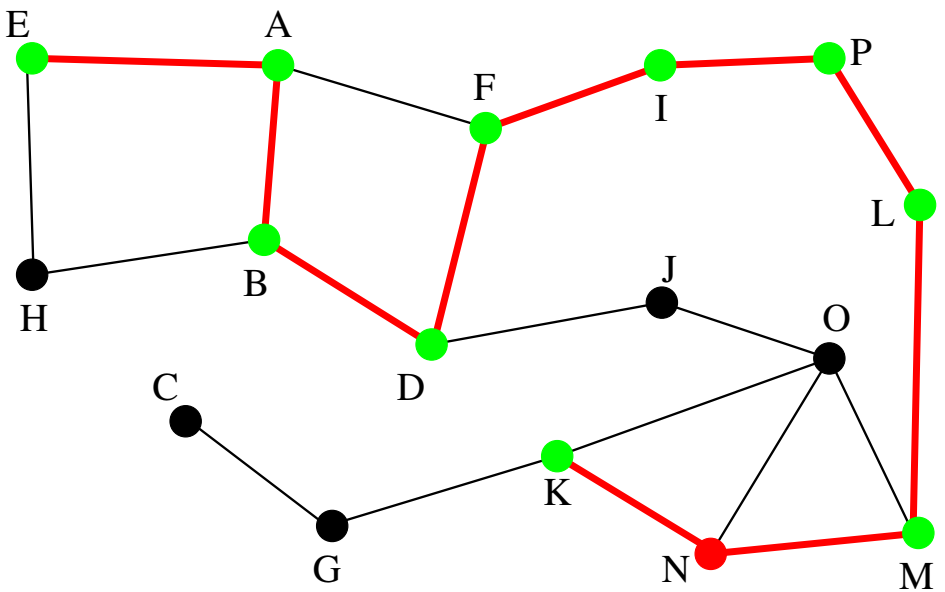
▸ Fin Animation



E	A	B	D	F	I	P	L	M	N					

Exemple

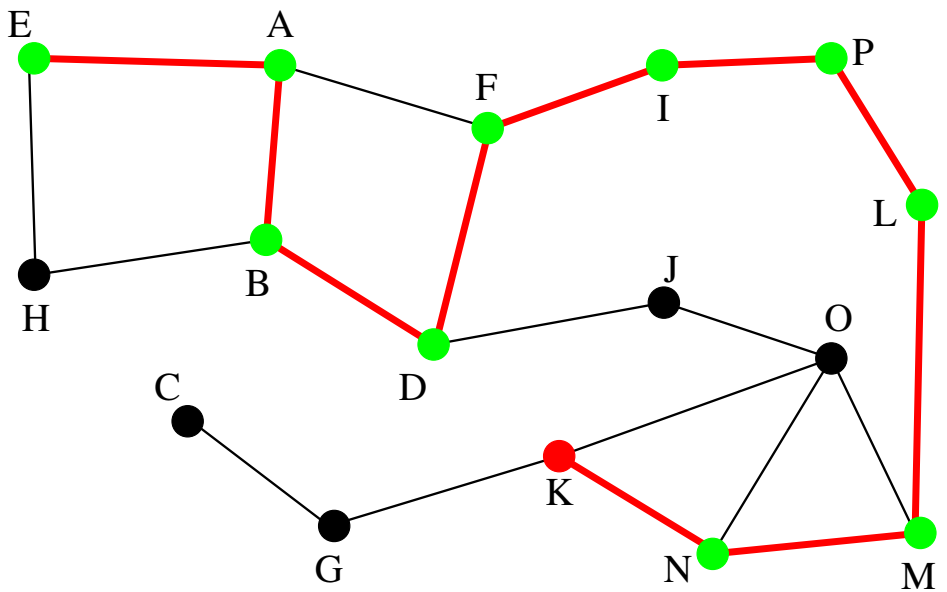
Fin Animation



E	A	B	D	F	I	P	L	M	N	K				

Exemple

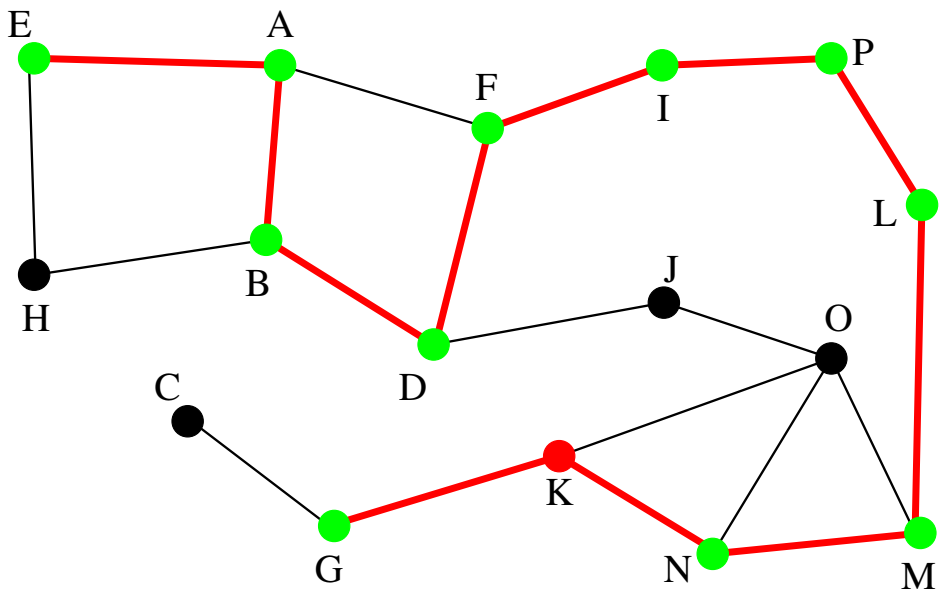
Fin Animation



E	A	B	D	F	I	P	L	M	N	K				

Exemple

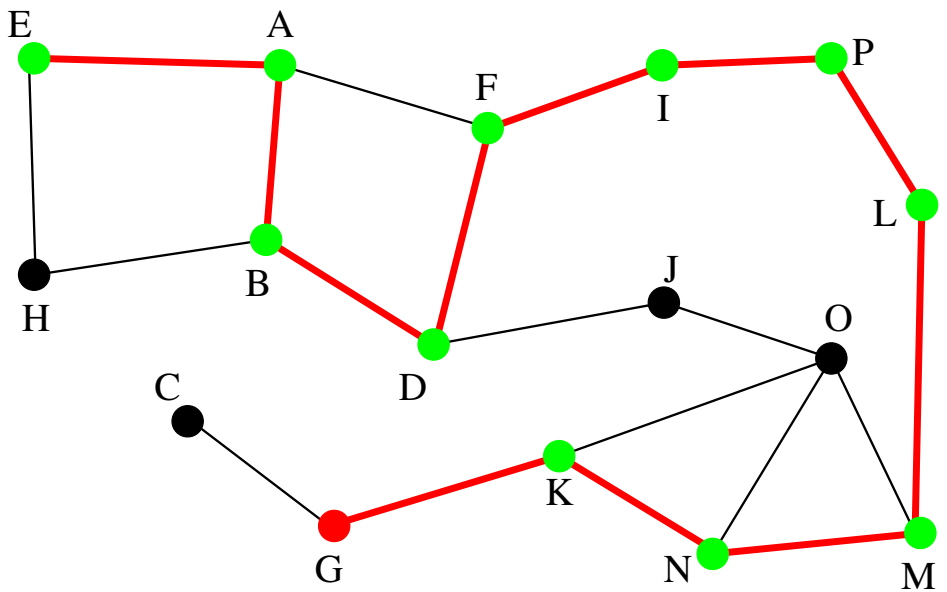
▸ Fin Animation



E	A	B	D	F	I	P	L	M	N	K	G			

Exemple

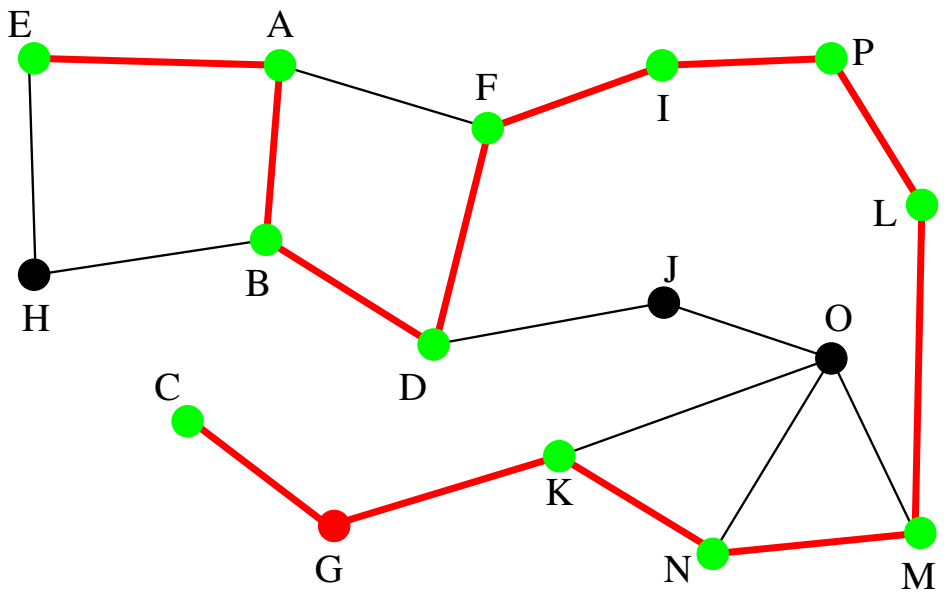
▸ Fin Animation



E	A	B	D	F	I	P	L	M	N	K	G			

Exemple

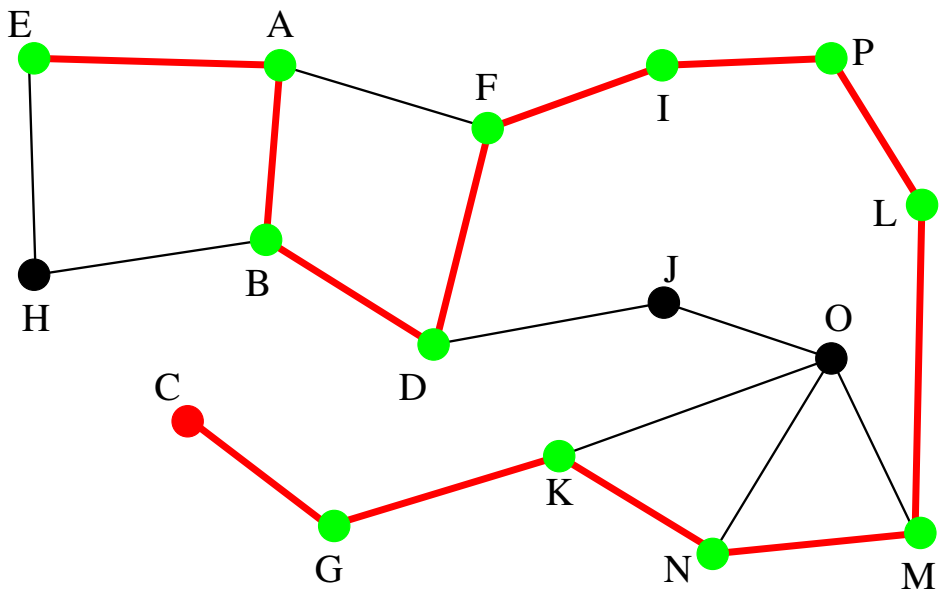
▸ Fin Animation



E	A	B	D	F	I	P	L	M	N	K	G	C		

Exemple

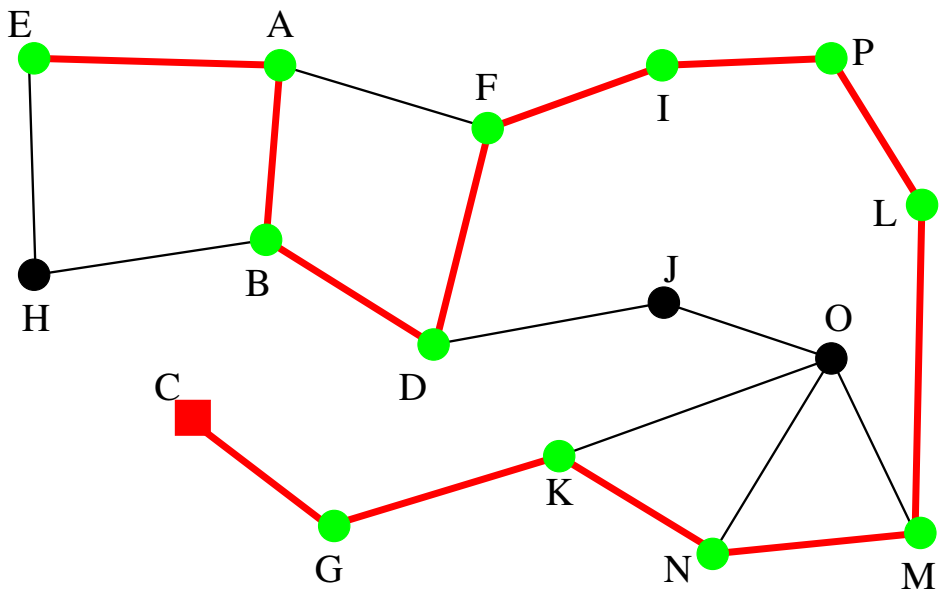
▸ Fin Animation



E	A	B	D	F	I	P	L	M	N	K	G	C		

Exemple

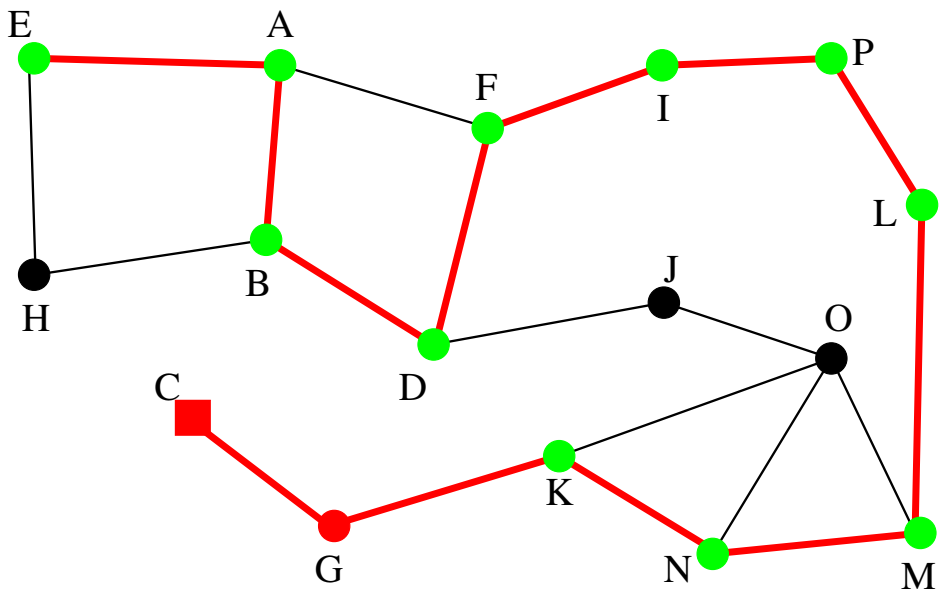
Fin Animation



E	A	B	D	F	I	P	L	M	N	K	G			

Exemple

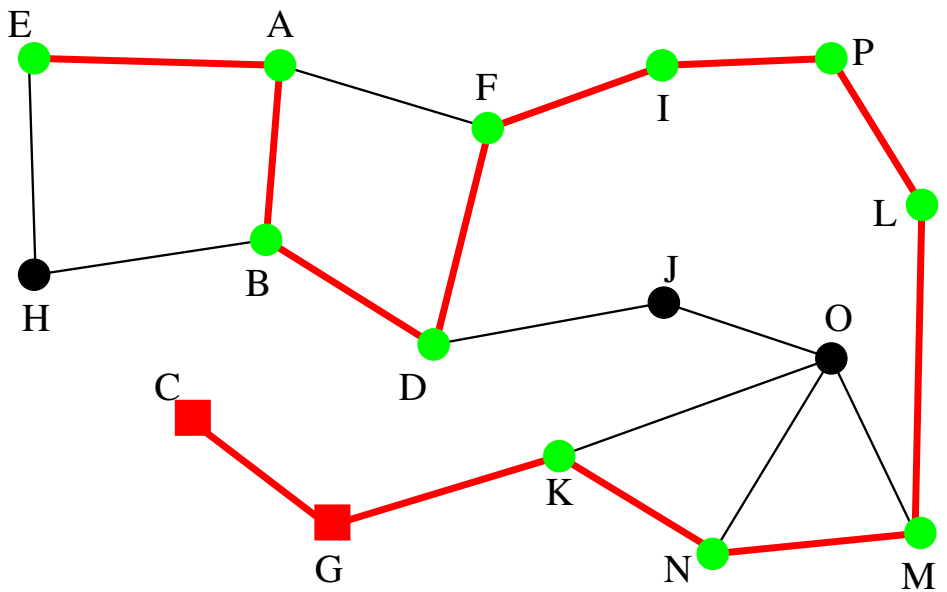
▸ Fin Animation



E	A	B	D	F	I	P	L	M	N	K	G			

Exemple

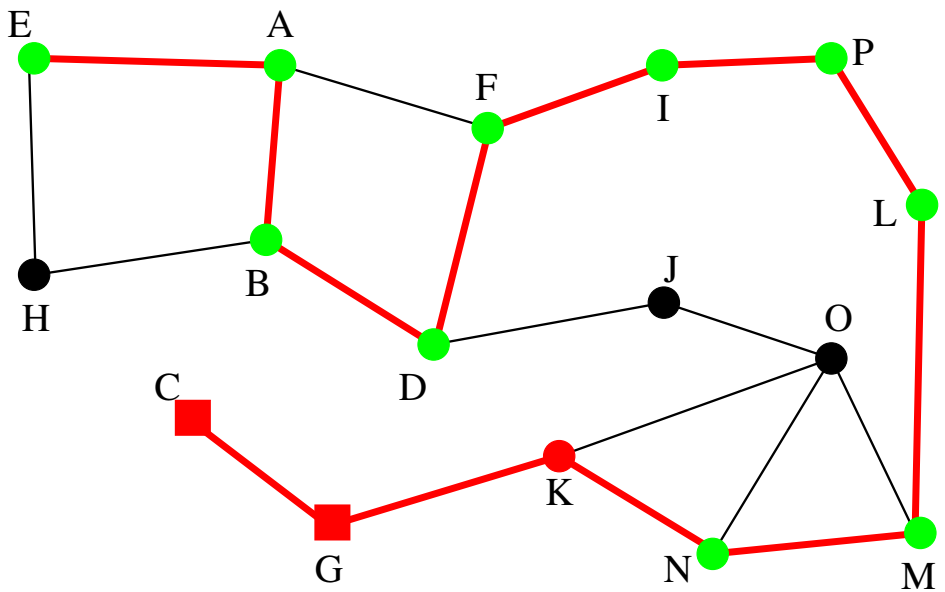
Fin Animation



E	A	B	D	F	I	P	L	M	N	K				

Exemple

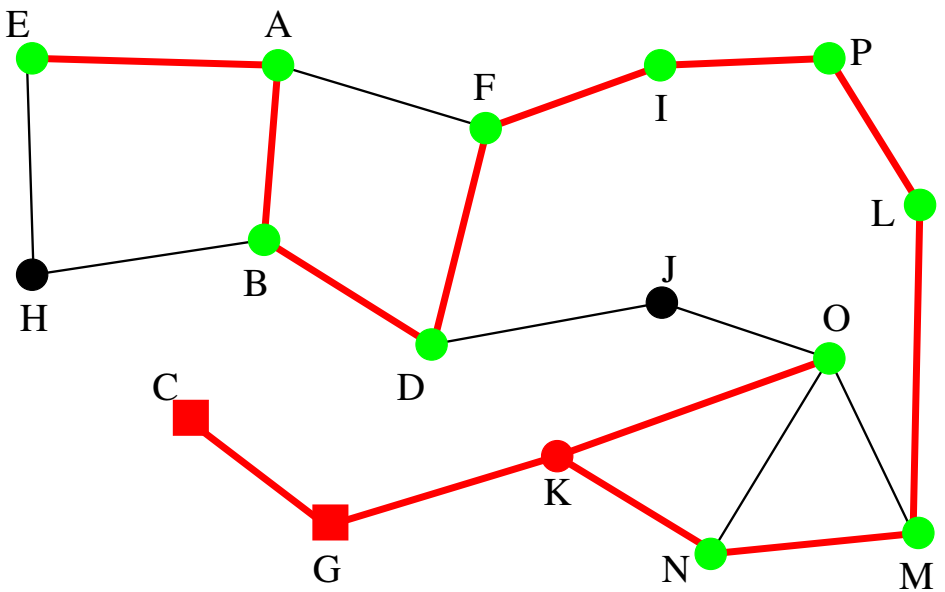
▸ Fin Animation



E	A	B	D	F	I	P	L	M	N	K				

Exemple

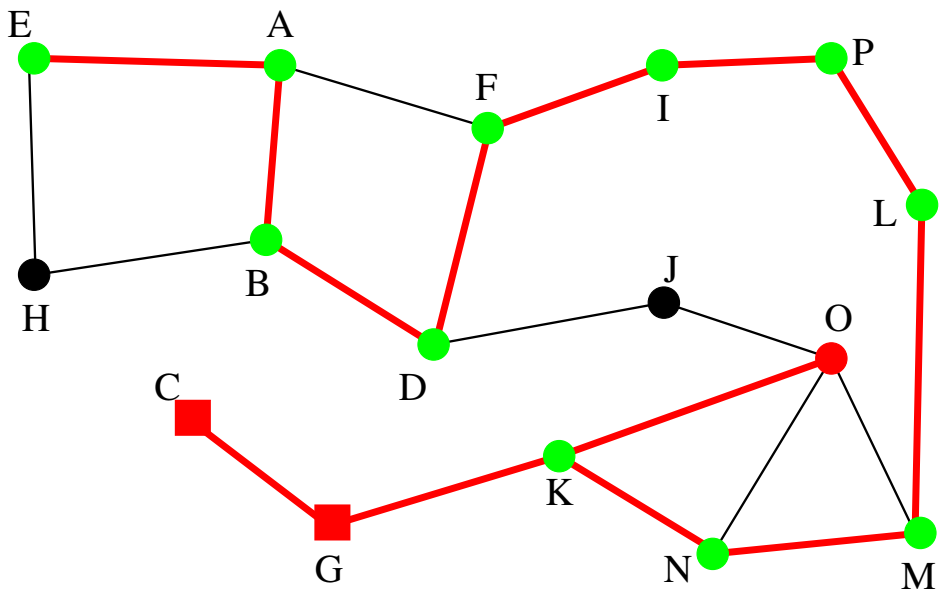
Fin Animation



E	A	B	D	F	I	P	L	M	N	K	O			

Exemple

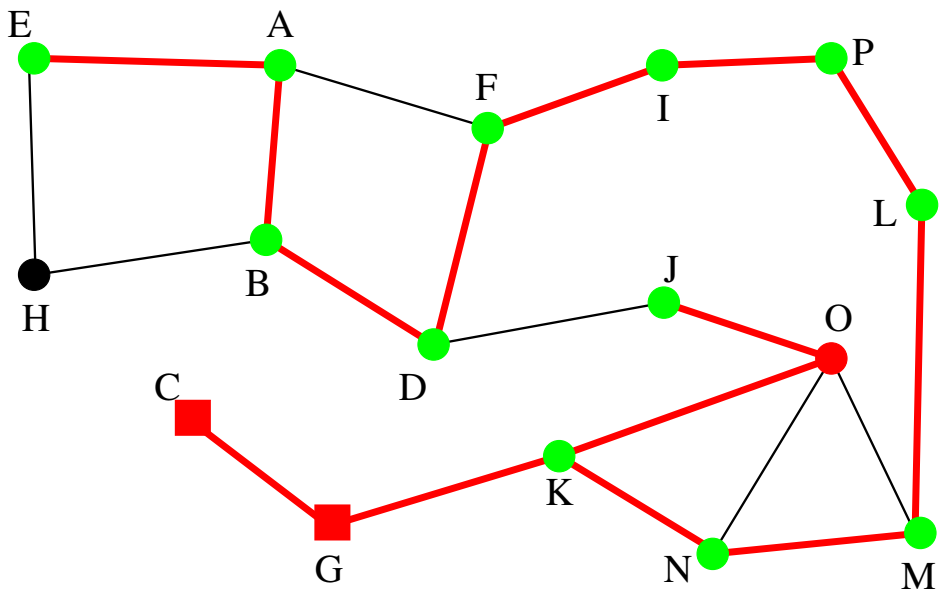
Fin Animation



E	A	B	D	F	I	P	L	M	N	K	O			

Exemple

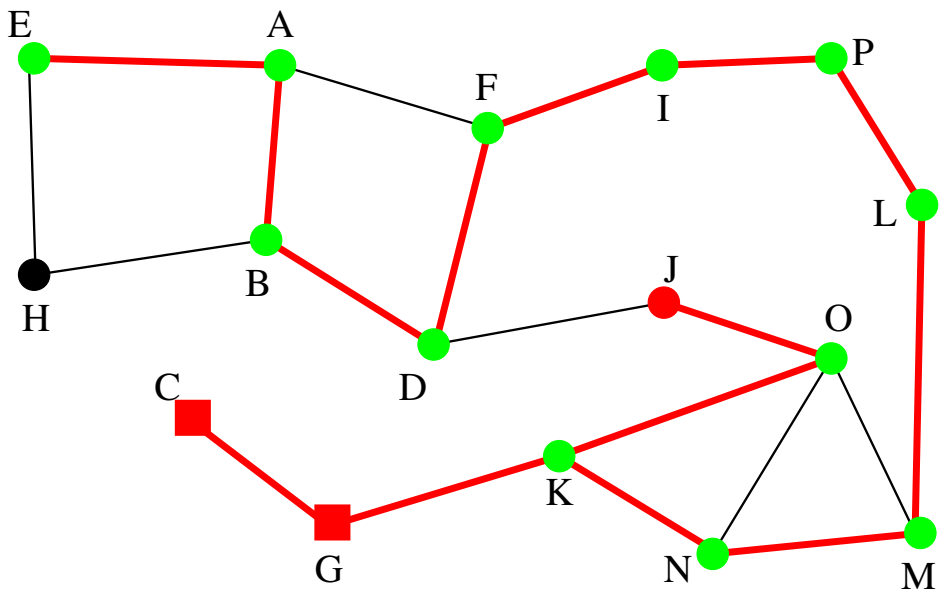
Fin Animation



E	A	B	D	F	I	P	L	M	N	K	O	J		

Exemple

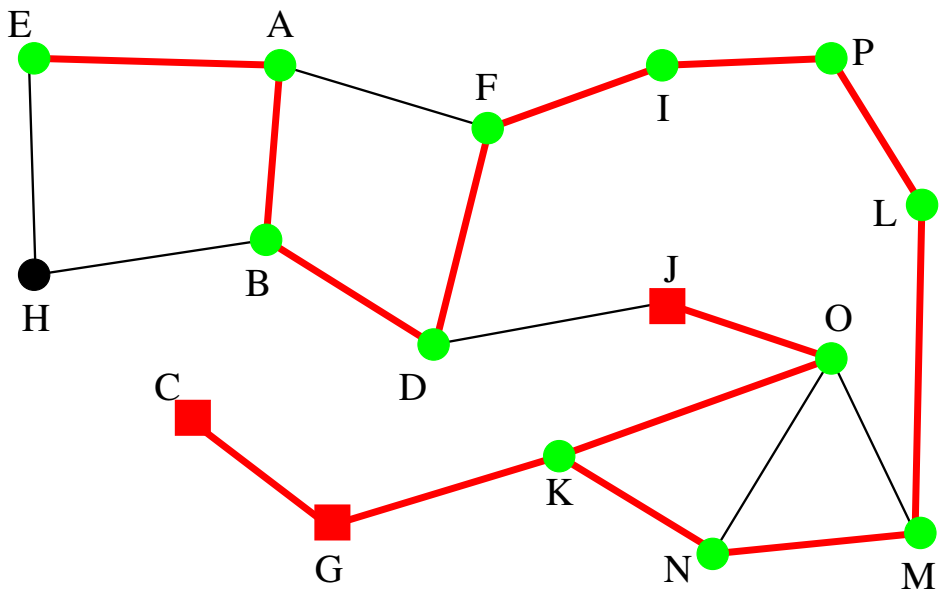
▸ Fin Animation



E	A	B	D	F	I	P	L	M	N	K	O	J		

Exemple

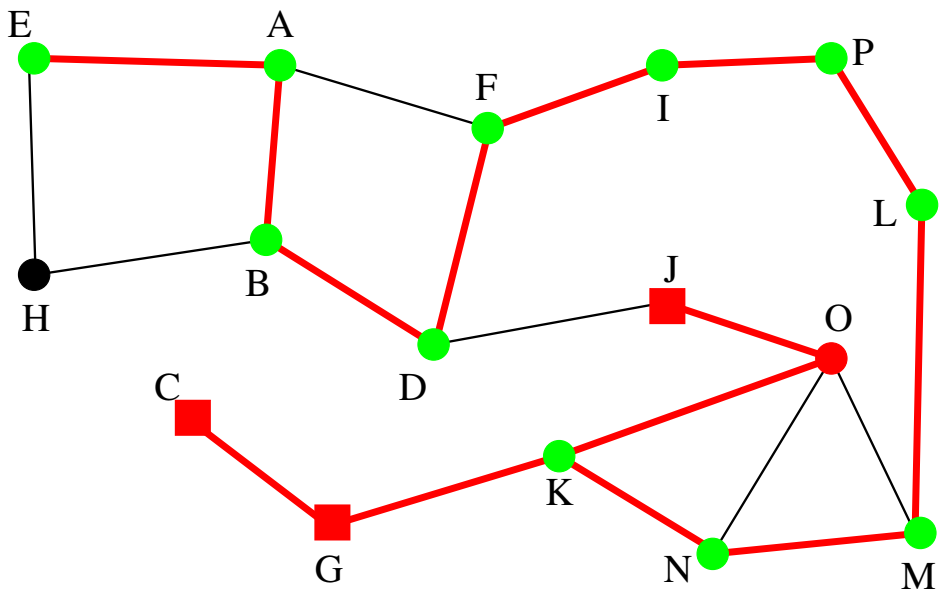
Fin Animation



E	A	B	D	F	I	P	L	M	N	K	O			

Exemple

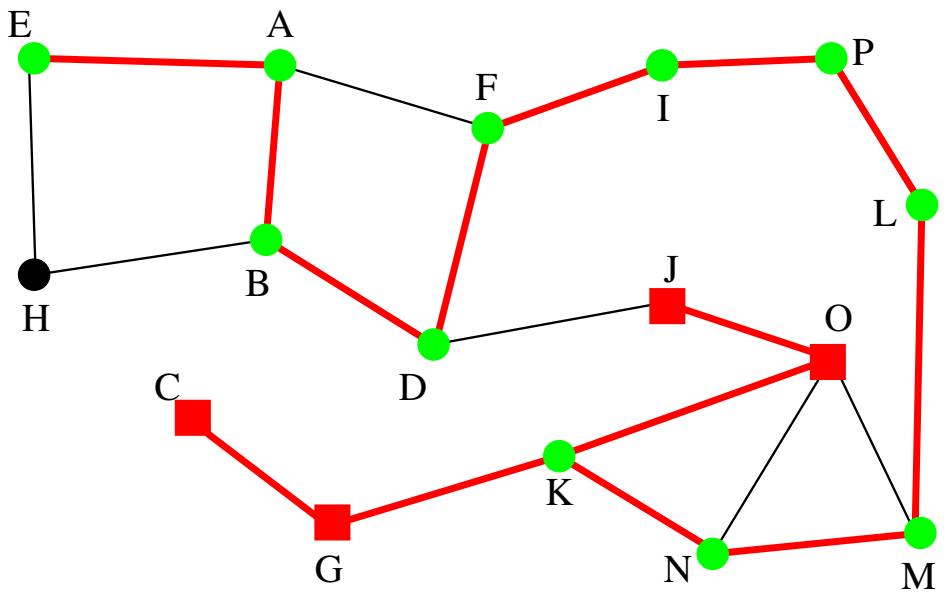
Fin Animation



E	A	B	D	F	I	P	L	M	N	K	O			

Exemple

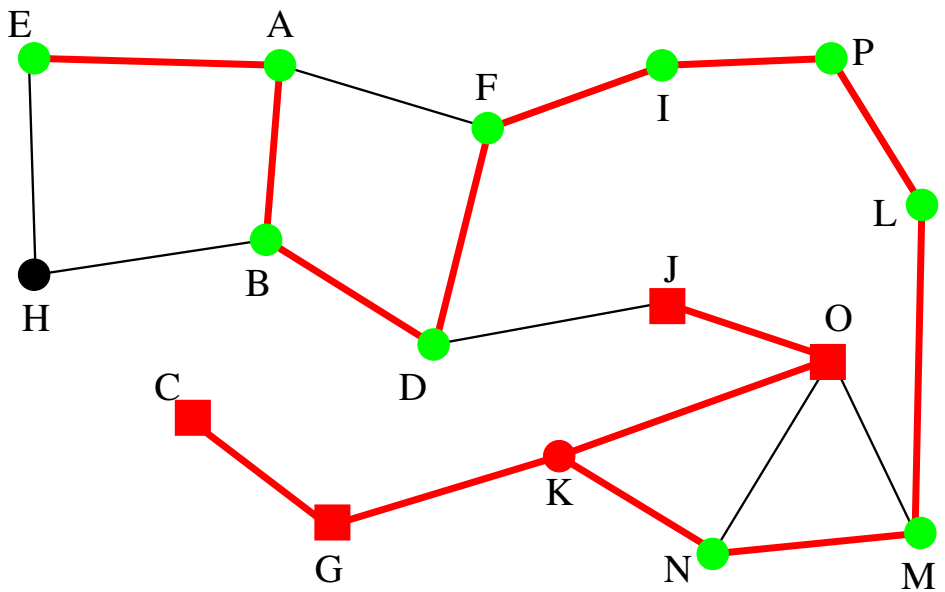
Fin Animation



E	A	B	D	F	I	P	L	M	N	K				

Exemple

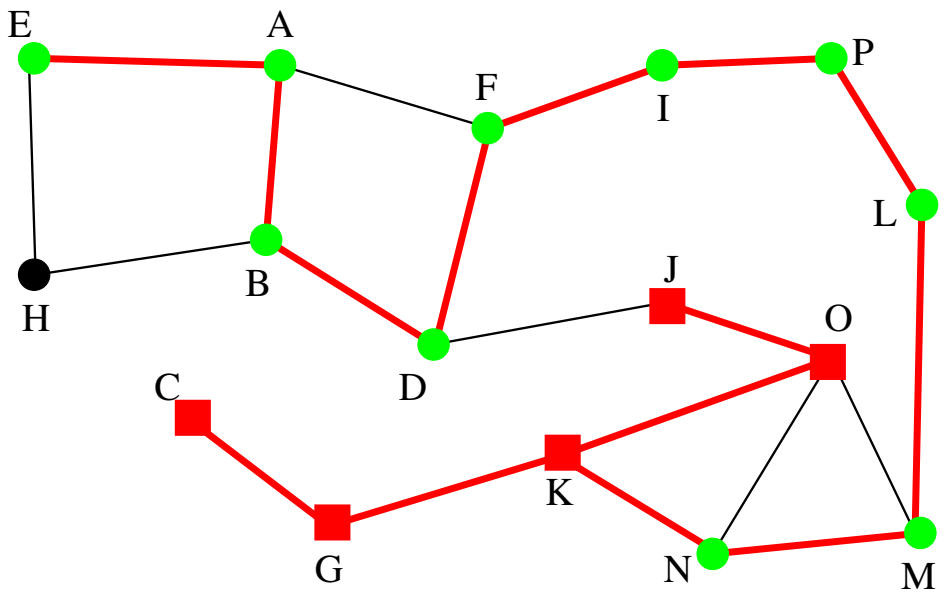
▸ Fin Animation



E	A	B	D	F	I	P	L	M	N	K				

Exemple

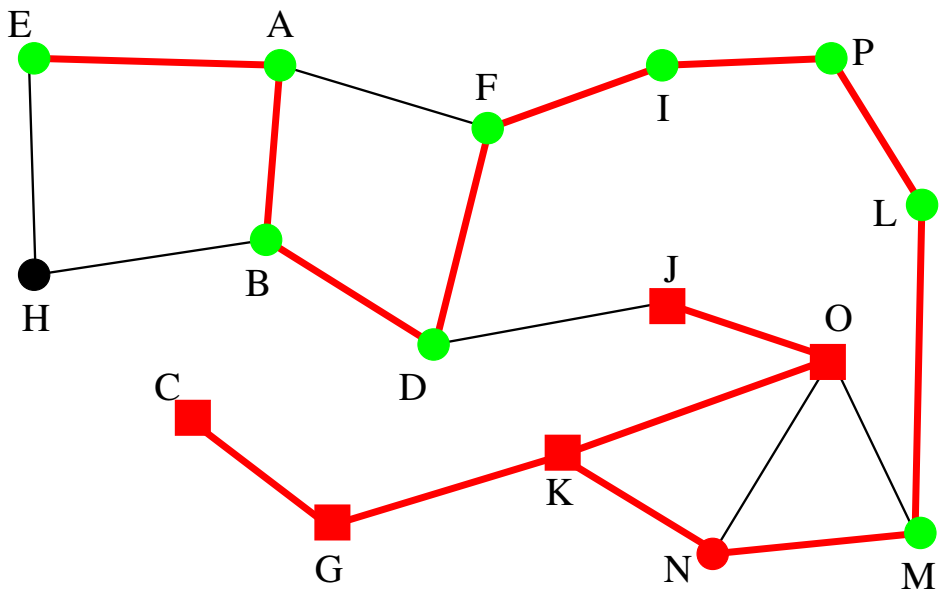
Fin Animation



E	A	B	D	F	I	P	L	M	N					

Exemple

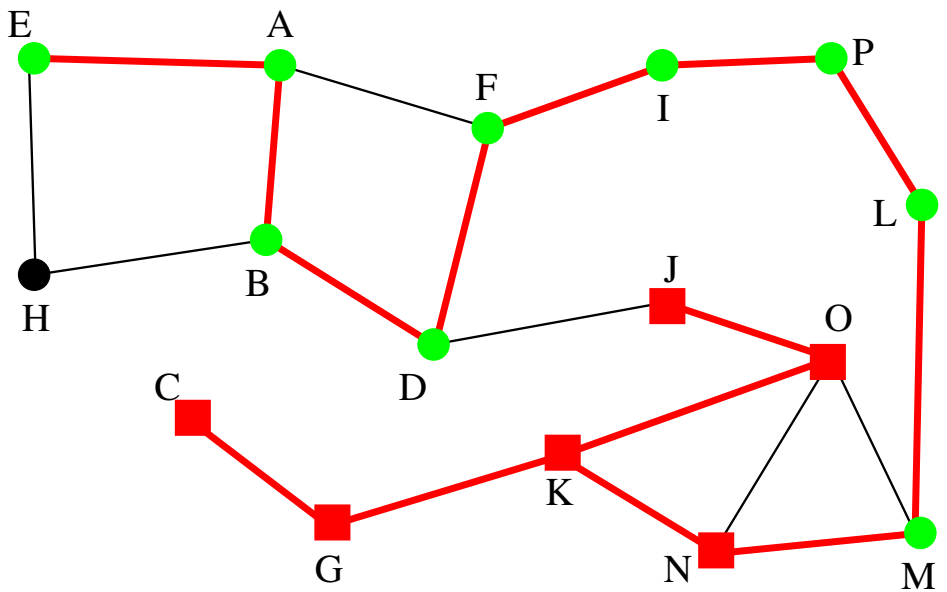
Fin Animation



E	A	B	D	F	I	P	L	M	N					

Exemple

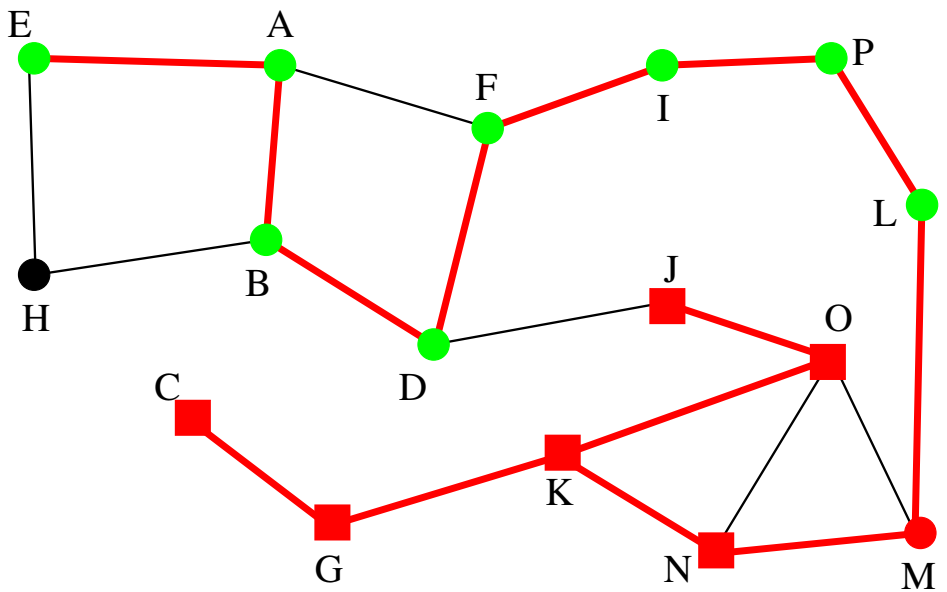
Fin Animation



E	A	B	D	F	I	P	L	M						

Exemple

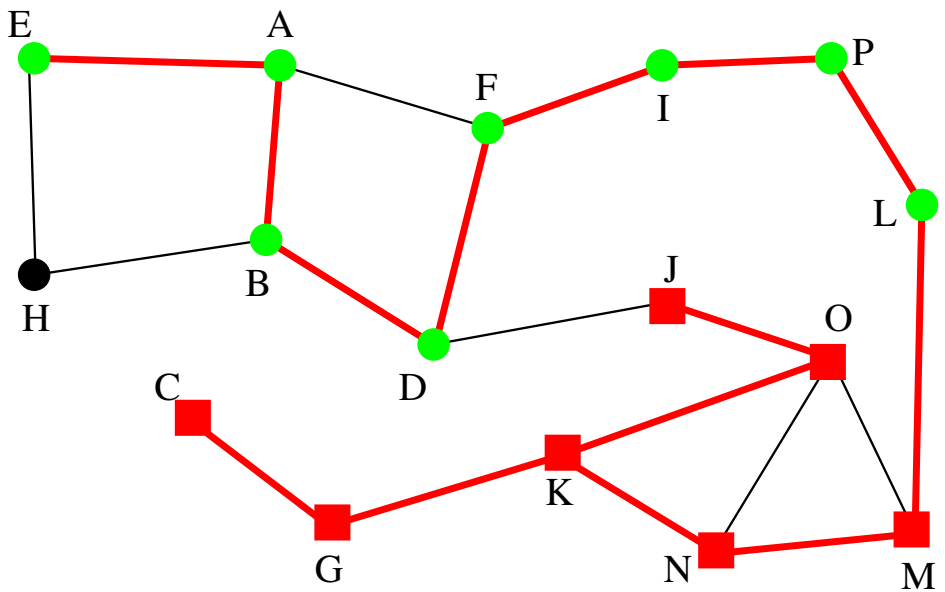
Fin Animation



E	A	B	D	F	I	P	L	M						

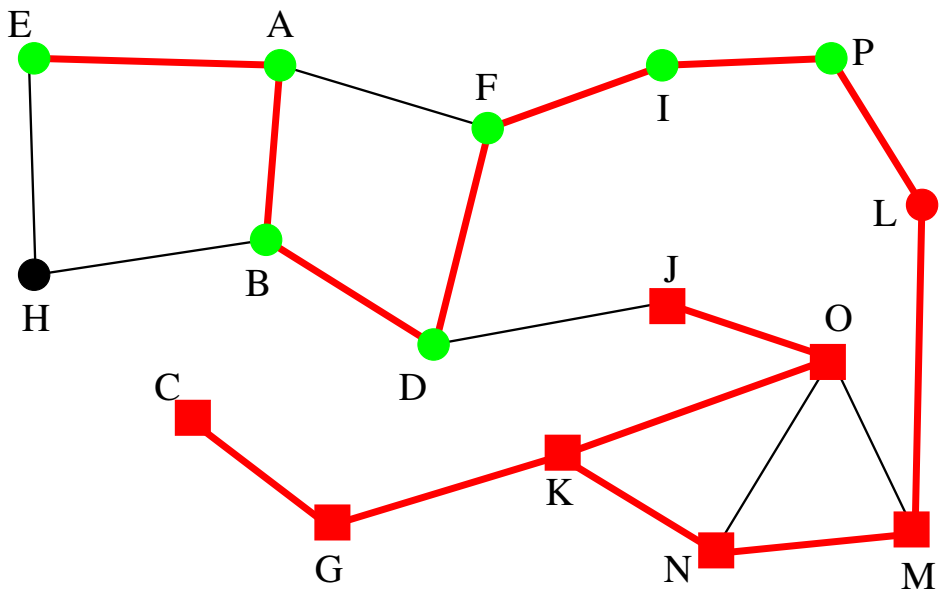
Exemple

▸ Fin Animation



Exemple

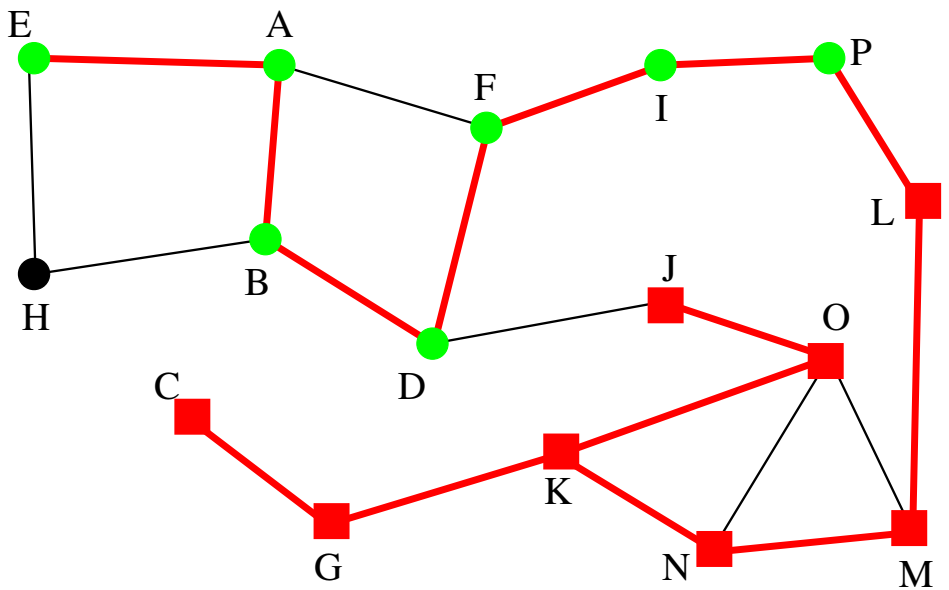
Fin Animation



E	A	B	D	F	I	P	L							

Exemple

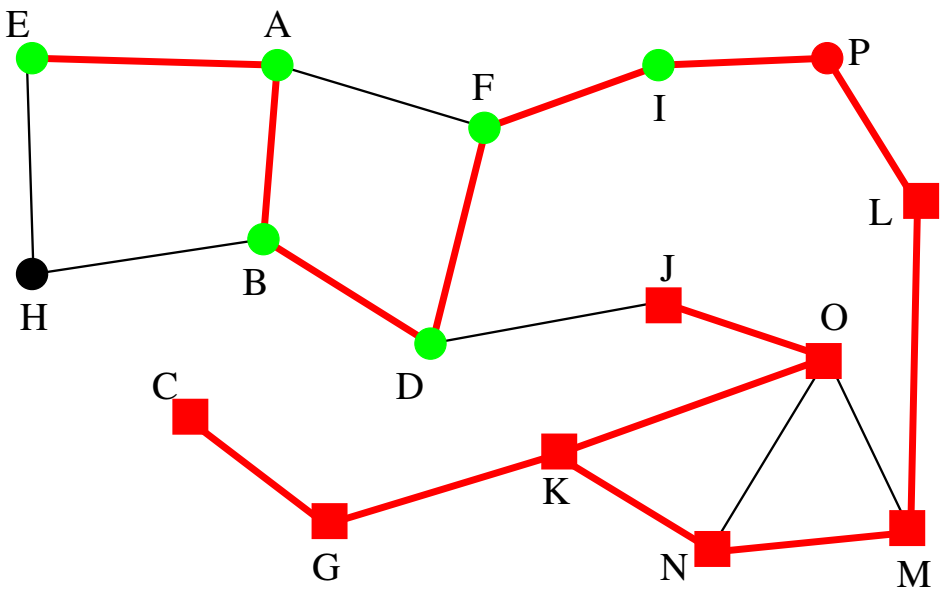
Fin Animation



E	A	B	D	F	I	P								

Exemple

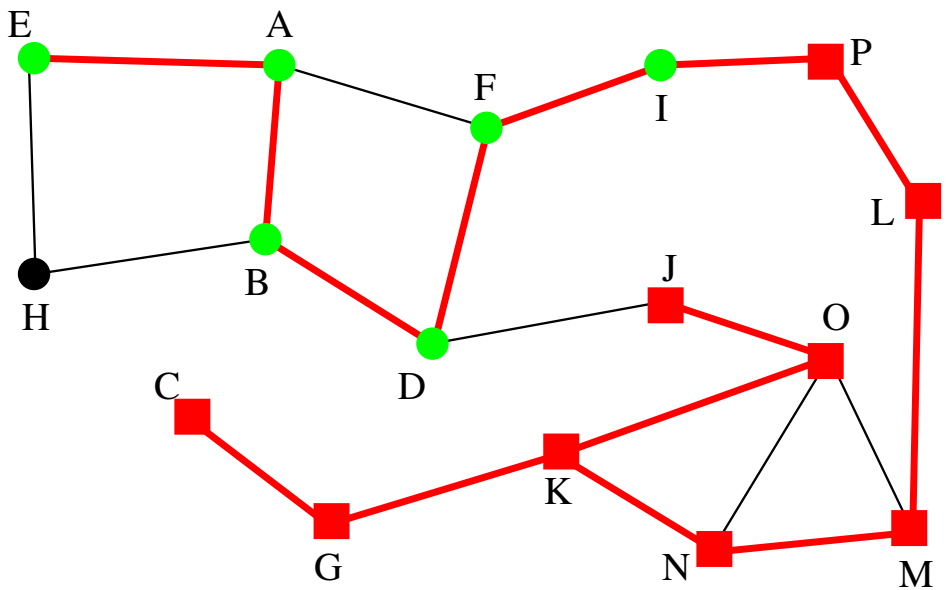
Fin Animation



E	A	B	D	F	I	P								

Exemple

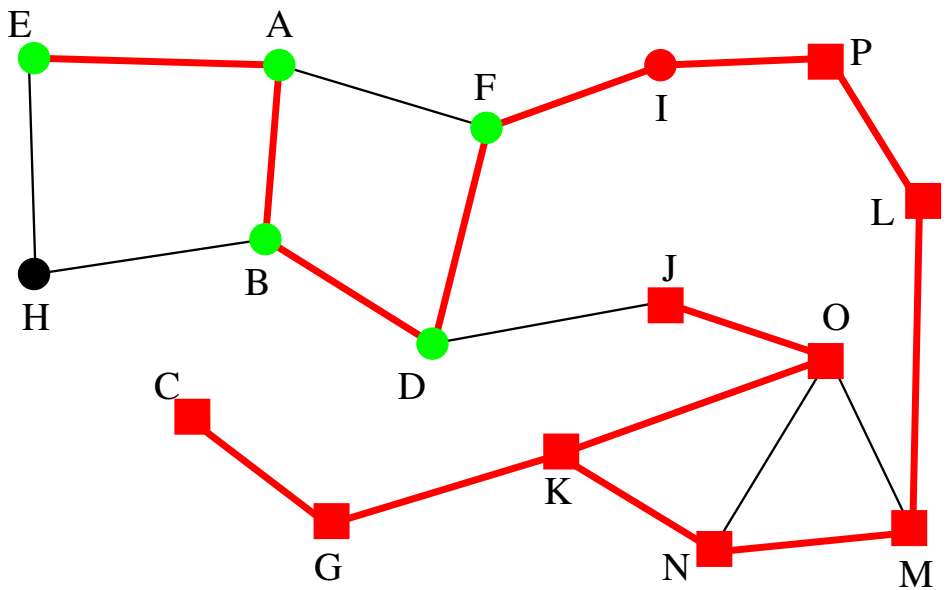
Fin Animation



E	A	B	D	F	I									

Exemple

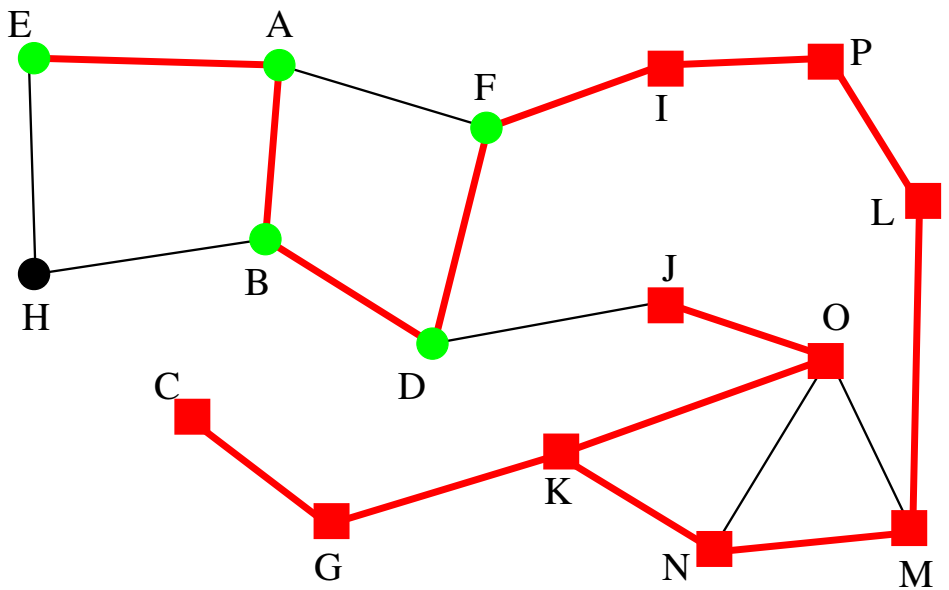
Fin Animation



E	A	B	D	F	I									

Exemple

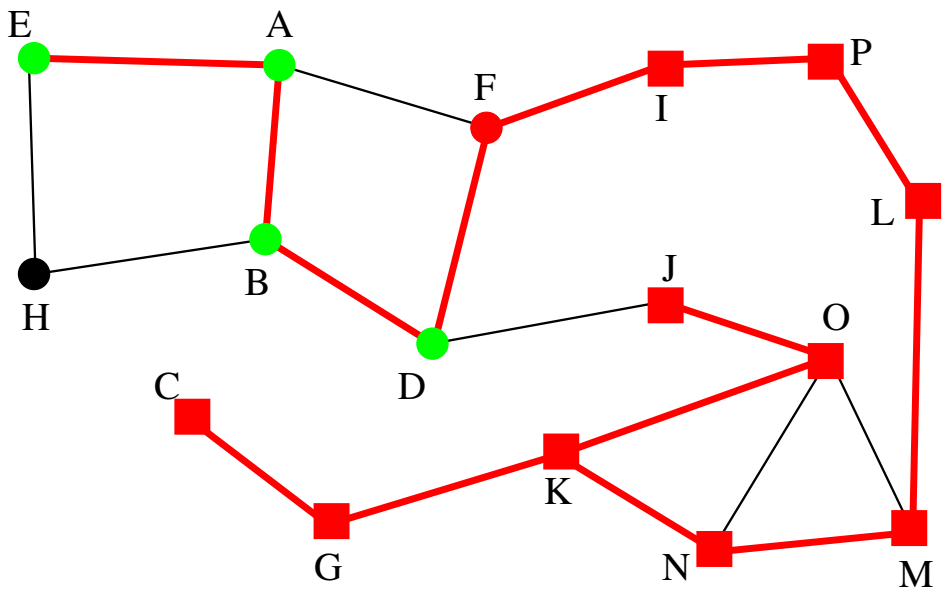
Fin Animation



E	A	B	D	F										

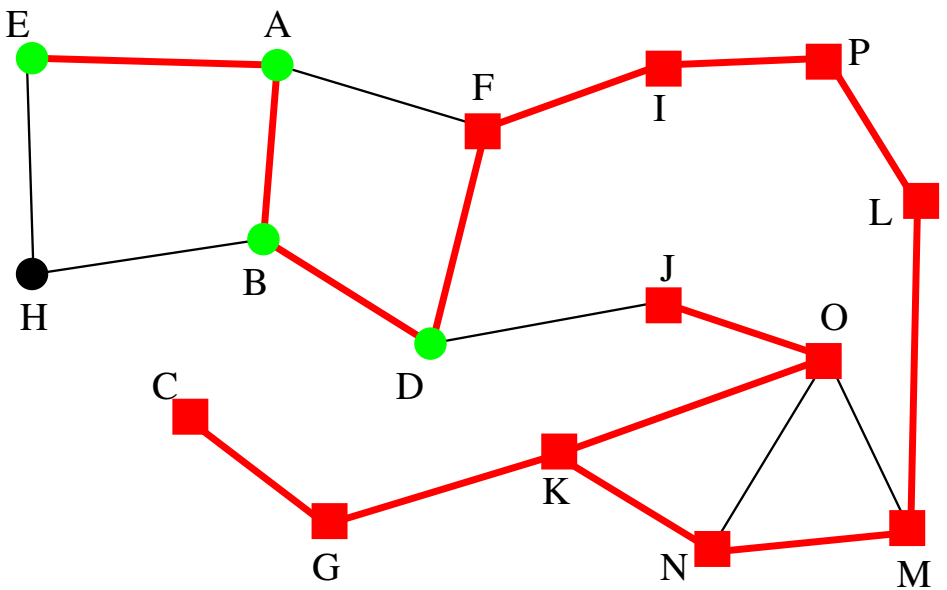
Exemple

Fin Animation



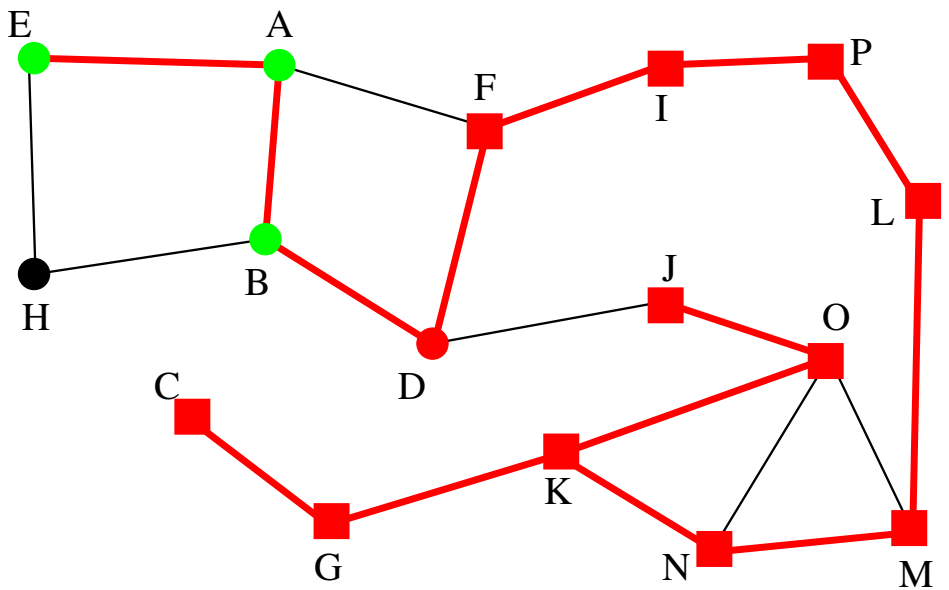
Exemple

Fin Animation



Exemple

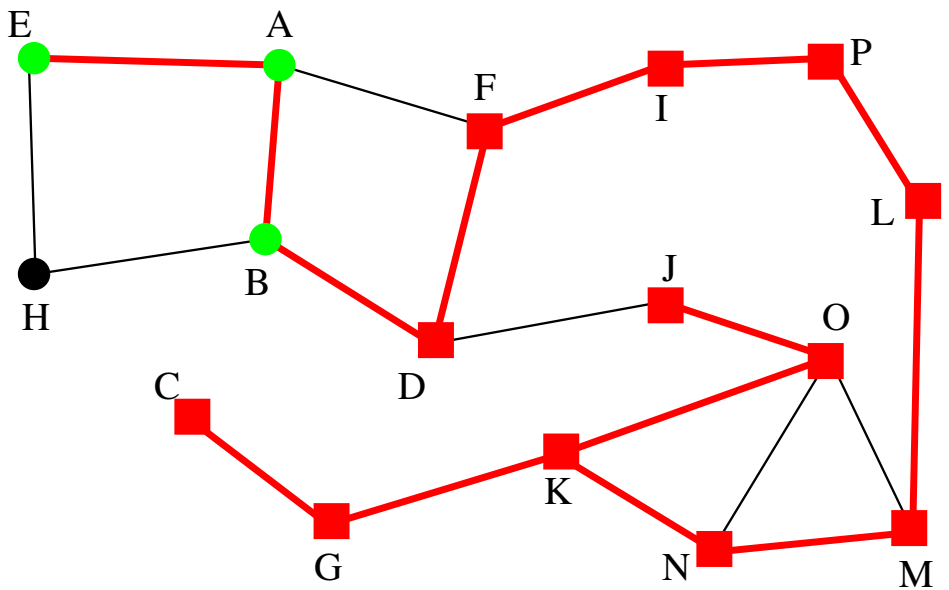
Fin Animation



E	A	B	D											

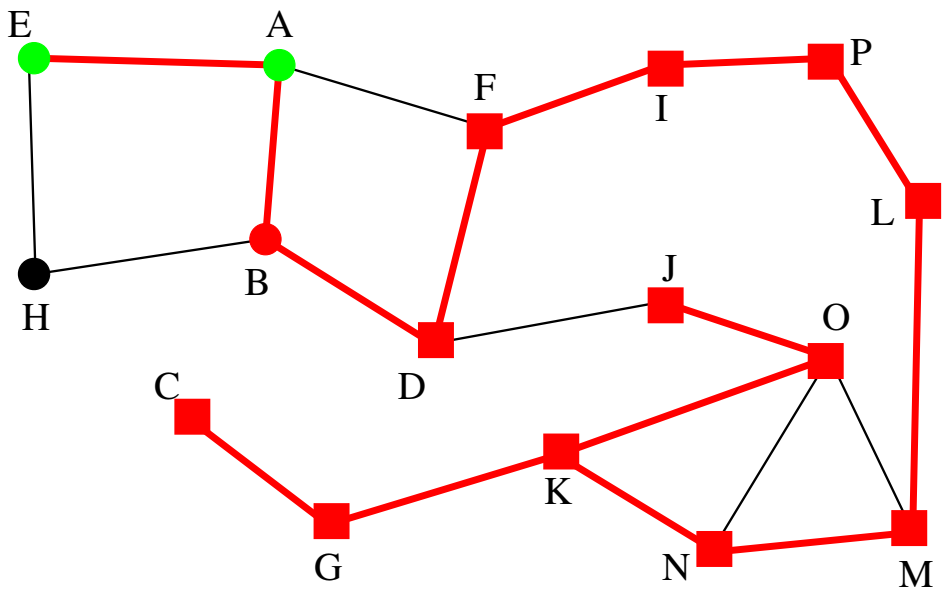
Exemple

Fin Animation



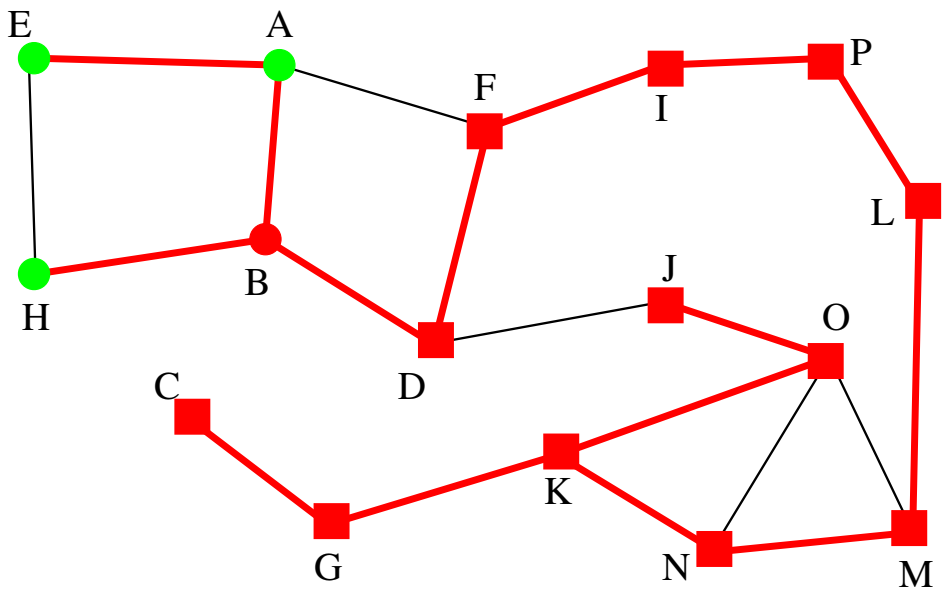
Exemple

Fin Animation



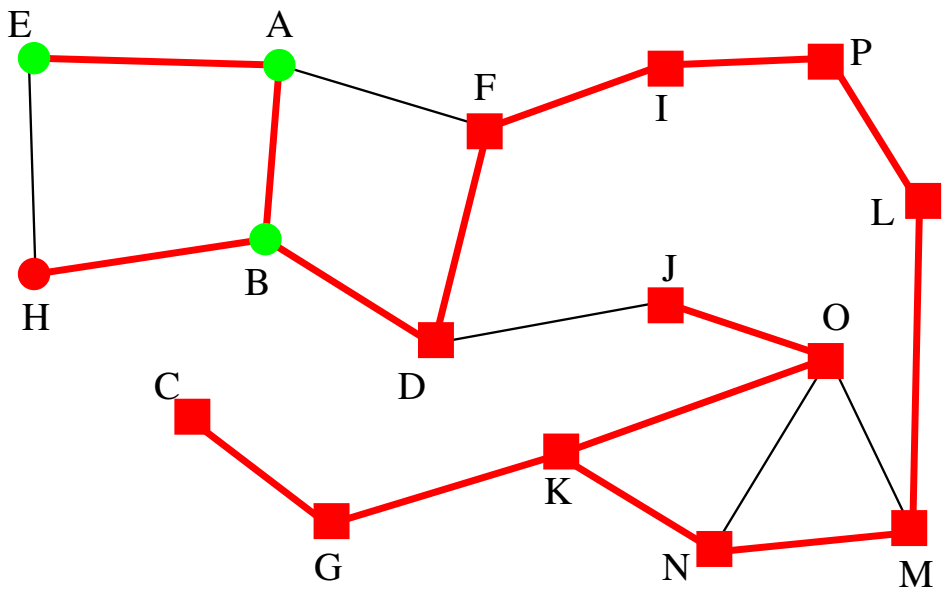
Exemple

Fin Animation



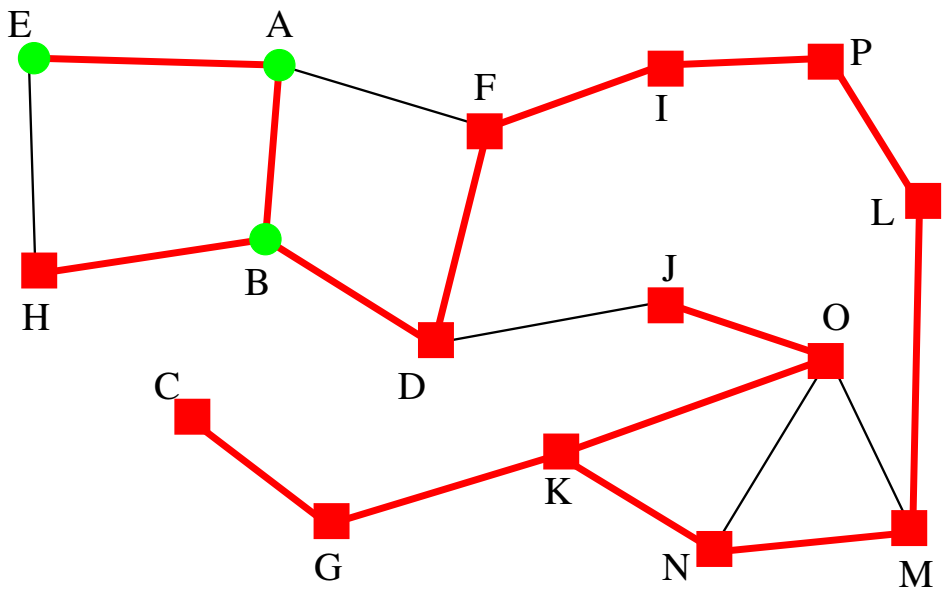
Exemple

Fin Animation



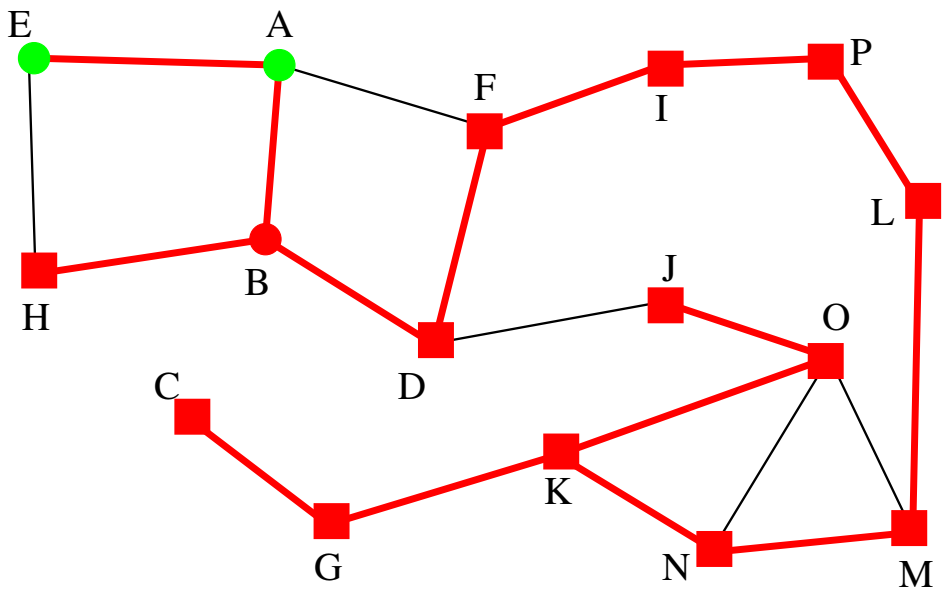
Exemple

Fin Animation



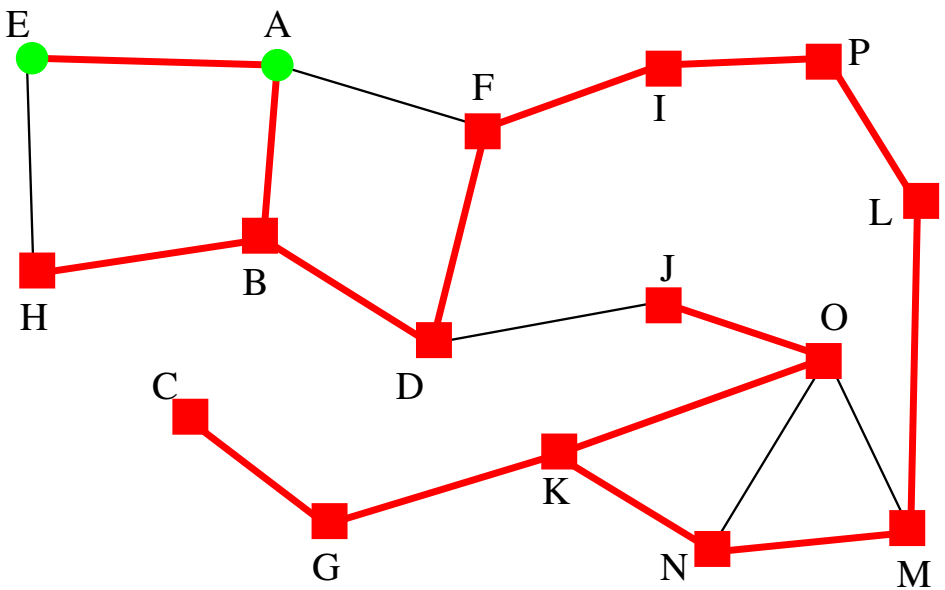
Exemple

Fin Animation



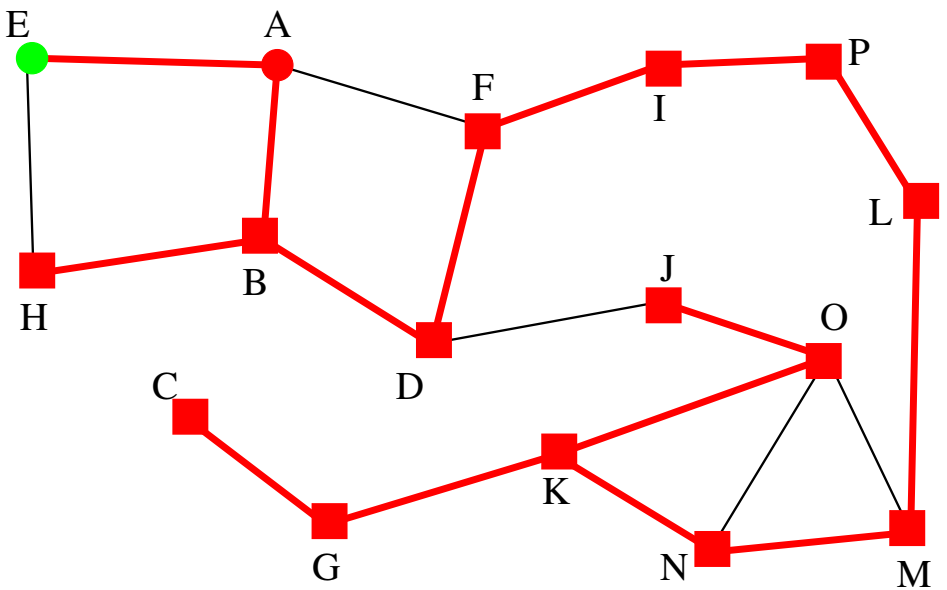
Exemple

Fin Animation



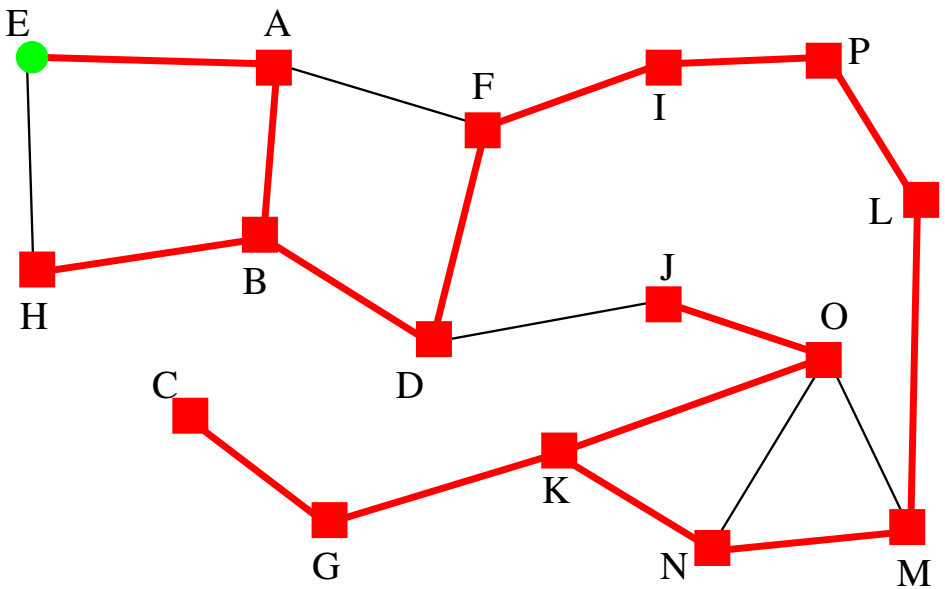
Exemple

Fin Animation



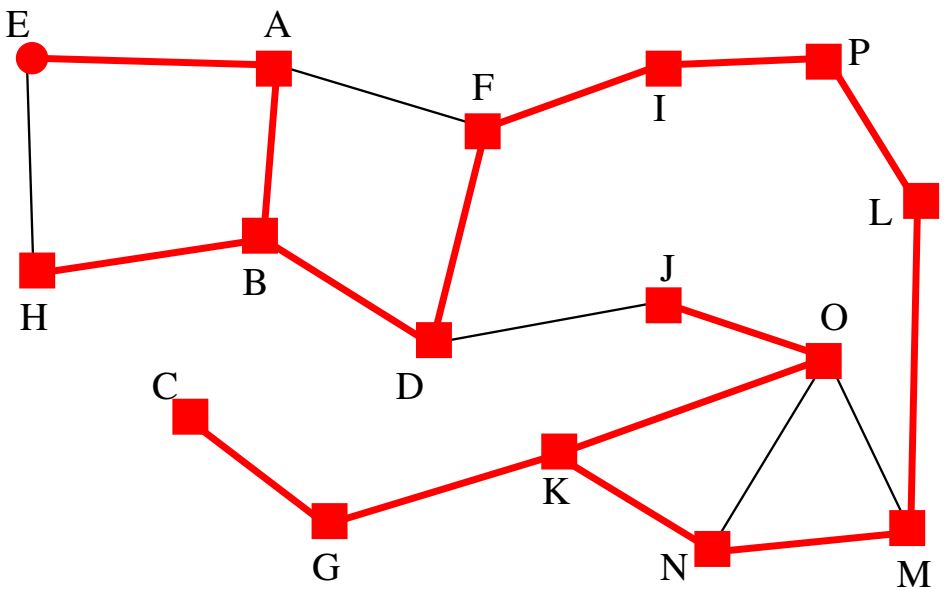
Exemple

Fin Animation



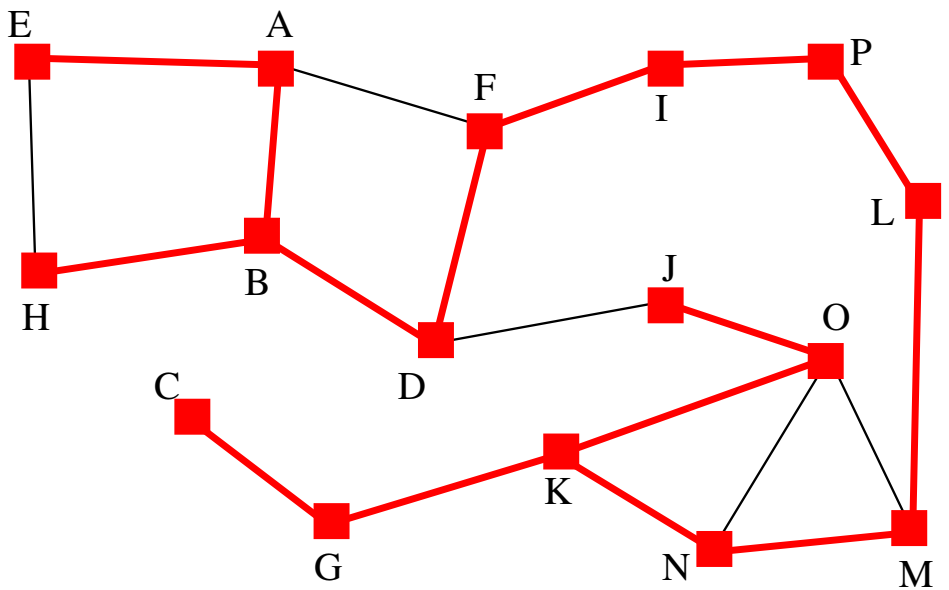
Exemple

Fin Animation



Exemple

Fin Animation



Algorithme du parcours en profondeur

Entrées

- G non orienté
- s source

Variables locales

- P pile (frontière)
- S ensemble des sommets connus
- u, v des sommets adjacents

Initialisation

- Ajouter s sur la pile
- Ajouter s à S

Tant que la pile n'est pas vide, répéter

- regarder le sommet v au dessus de la pile
- si v n'a pas de voisin inconnu
  - alors enlever v de la pile
  - sinon pour chaque voisin inconnu u de v ajouter u au dessus de la pile (devant v) et ajouter u à S

Algorithme 3 : Parcours en profondeur

Données *G un graphe et s un sommet de G*
Variables locales
S un ensemble `/* zone connue */`
P une pile `/* la frontière */`
u, v deux sommets
début
  initialisation
  `ajouter(`s`,`S`)`
  `empiler(`s`,`P`)`
  répéter
    $v :=$`valeurSommetPile (`P`)`
    si *il existe* $u \notin S$ *adjacent à* v **alors**
      `ajouter(`u`,`S`)` `/* première visite de` u `*/`
      `empiler(`u`,`P`)`
    sinon
      `depiler(`P`)` `/* dernière visite de` v `*/`
    fin si
  jusqu'à `estVide(`P`)`
fin

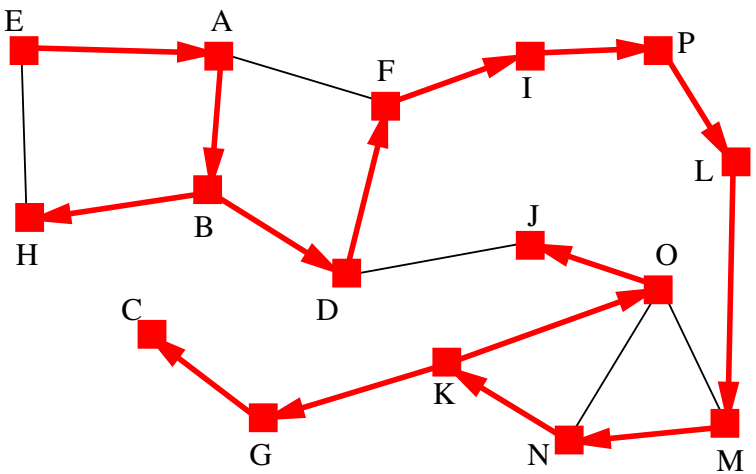
Sommaire

1. Introduction
2. Exemple
3. File et pile
4. Parcours en largeur
   - Rappel sur les files
   - Algorithme
   - Arbre de parcours et calcul des distances
5. Parcours en profondeur
   - Rappel sur les piles
   - Algorithme
   - Arbre de parcours
6. Pour finir

Propriétés d'un arbre de parcours

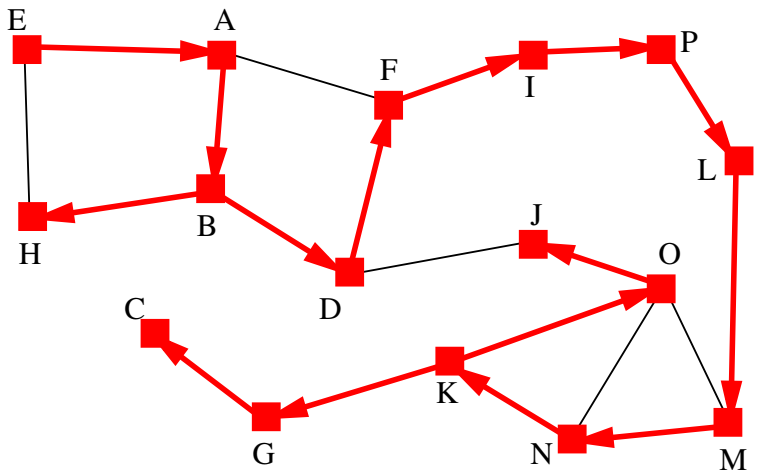
Remarque

Dans le cas d'un arbre de parcours, on part de la source. On a donc un arbre enraciné. Cela revient à considérer un arbre orienté par le sens du parcours depuis la source vers les feuilles. On peut donc parler d'ancêtre d'un sommet.



exemple pour le parcours en profondeur (source = E)

Propriétés d'un arbre de parcours en profondeur



Théorème 2 (absence d'arc transverse)

Soit $\{s, t\}$ une arête du graphe sous-jacent G. Alors, l'un des deux sommets est toujours l'ancêtre de l'autre dans l'arbre de parcours en profondeur.

Exemples : A ancêtre de F, D ancêtre de J, N ancêtre de K, etc.

Sommaire

1. Introduction
2. Exemple
3. File et pile
4. Parcours en largeur
   - Rappel sur les files
   - Algorithme
   - Arbre de parcours et calcul des distances
5. Parcours en profondeur
   - Rappel sur les piles
   - Algorithme
   - Arbre de parcours
6. Pour finir

Conclusion

- Un parcours de graphe connexe induit un arbre couvrant de ce dernier, dit arbre de parcours.
- La manière dont on stocke/gère la frontière lors du parcours change drastiquement le comportement du parcours et l'arbre couvrant obtenu :
  - Avec une file, on obtient un arbre de parcours en largeur.
  - Avec une pile, on obtient un arbre de parcours en profondeur.
- Les 2 méthodes fonctionnent aussi pour les graphes orientés.
- Chaque méthode a de bonnes propriétés qu'on peut exploiter.
  - Le parcours en largeur permet de déduire les distances entre la source et les autres sommets.
  - Le parcours en profondeur permet de calculer un ordre topologique (sera vu plus tard).