- **lemme (fondamental)** : tout DAG possède une source.

Pr : Depuis un sommet x_1 quelconque, on construit un chemin maximal dirigé vers x_1 :

$$x_1 \leftarrow x_2 \leftarrow x_3 \dots \leftarrow x_k$$

x_k n'a pas de voisin entrant dans $\{x_1, \dots x_{k-1}\}$ sinon, on a un circuit. Donc x_k est une source. ∎

- De même, tout DAG possède un puit.

- Un tri topologique d'un graphe orienté est une énumération $v_1, v_2 \dots v_n$ de ses sommets telle que pour tout arc $v_i v_j$ de D, on a : $i < j$. (Les arcs vont de la "gauche vers la droite").

- **Proposition** : D est un DAG ssi il admet un tri topologique.

Pr : $\Leftarrow$: Si D a un tri topologique, il ne peut pas avoir de circuit.

$\Rightarrow$: D est un DAG $\rightarrow$ il a une source v_1

$D \setminus v_1$ ———— $\rightarrow$ ———— v_2

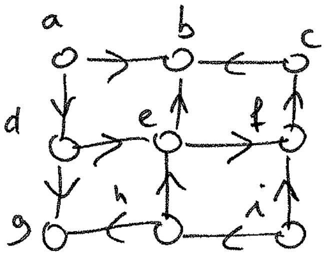
$D \setminus \{v_1, v_2\}$ ———— $\rightarrow$ ———— v_3

⋮

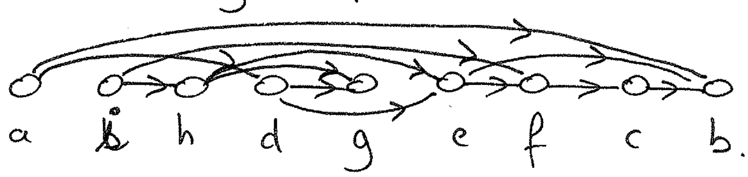
$\downarrow$ tri topologique.

∎

Exemple :



a i h d g e f c b.



- L'algo donné ici est en $O(n^2)$ ($O(n)$ pour trouver une source, répété n fois). On verra en TD un algo en $O(n+m)$

qui utilise un parcours en profondeur.

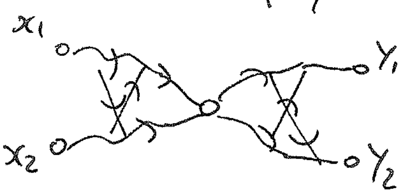
- Problème du jour n°6 -

2-CHEMIN-DISJOINTS $(D; \{x_1, x_2\}; \{y_1, y_2\})$.

ENTRÉE : D graphe orienté, 2 sommets départs $\{x_1, x_2\}$, 2 sommets arrivés $\{y_1, y_2\}$.

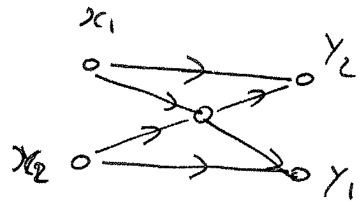
SORTIE : 2 chemins disjoints partant de $\{x_1, x_2\}$ terminant en $\{y_1, y_2\}$

- Il existe un algo polynomial (Flots), en fait l'obstruction est :



un sommet séparateur "entre $\{x_1, x_2\}$ et $\{y_1, y_2\}$" (voir Master 1).

- Peut-on spécifier un chemin de x_1 à x_2 et un chemin de y_1 à y_2 ! c'est plus dur, par exemple :



2-CHEMINS-DISJOINTS-AVEC-EXTRÉMITÉS-FIXÉES $(D; (x_1, x_2); (y_1, y_2))$

ENTRÉE : D orienté, $x_1, x_2, y_1, y_2 \in V(D)$.

SORTIE : 2 chemins disjoints : un de x_1 à y_1, un de x_2 à y_2

- NP-difficile ! (Fortune, Hopcroft et Wyllie - 1980)
- Et dans les graphes non orientés ? → Ça devient polynomial (Seymour Rob. 95)

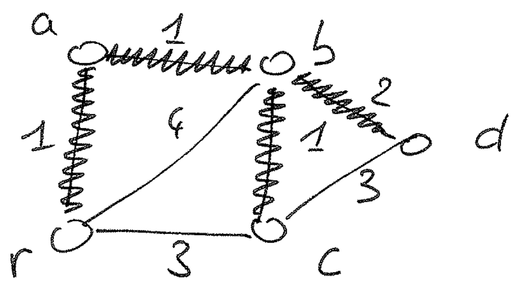
Chap V : Plus courts chemins

I) Graphes valués

- On cherche un arbre des plus courts chemins, mais cette fois, il y a des longueurs sur les arêtes :

  Entrée : $G=(V,E)$, $l : E \to \mathbb{R}^+$ les longueurs, $r \in V$ la racine.
  Sortie : Un arbre des plus courts chemins enraciné en r.

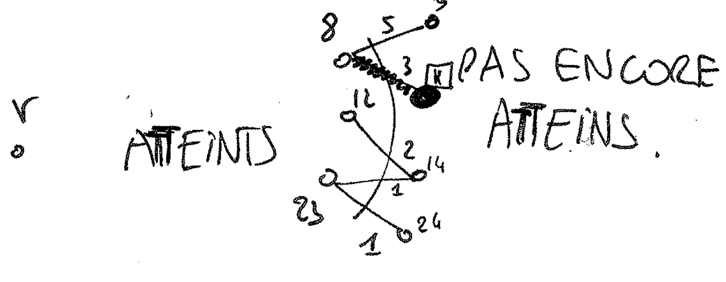
Exemple :



Pour chaque sommet x, le chemin dans l'arbre de x à r est un p.c.c de x à r

- **Algo Manuel** :
  - Clouez r au mur
  - Remplacez les autres sommets par des perles.
  - Remplacez chaque arête xy par un fil de longueur $l(xy)$.
  - lâchez tout !

- **Dijkstra** (1959) : Pour chaque sommet x on calcule sa distance à r : $d(x)$.



On maintient un ensemble de sommets traités.

A chaque étape, on traite le sommet pas encore traité qui sera le moins loin de r.

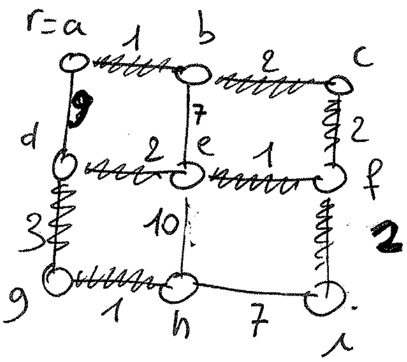
DIJKSTRA (G par liste de Voisins, r).

Initialisation

Pour tout $v \in V$
 $d(v) \leftarrow +\infty$;
 traité(v) $\leftarrow 0$;
père(r) $\leftarrow$ r;
~~traité(r) $\leftarrow 1$;~~
$d(r) \leftarrow 0$;

Tant qu'il existe x avec traité(x) = 0
 Choisir un tel x avec d(x) minimum
 traité(x) $\leftarrow 1$;
 Pour tout $y \in$ Vois(x)
  Si traité(y) = 0 et $d(y) > d(x) + \ell(xy)$ // On teste si x est un raccourci. Si oui, on relaxe l'arête xy.
   $d(y) \leftarrow d(x) + \ell(xy)$;
   père(y) $\leftarrow$ x;

• Exemple :



	a	b	c	d	e	f	g	h	i
père	a	a	b	~~a~~ e	~~b~~ f	c	d	~~e~~ ~~i~~ g	f
d	0	1	3	~~9~~ 8	~~8~~ 6	5	~~11~~ 10	~~16~~ ~~14~~ 11	7

Analyse :

- **Terminaison** : à chaque passage dans le Tant que (L_7) le nombre de sommet à traité$(x) = 1$ croît strictement → OK.

- **Complexité** : Init → $O(n)$.

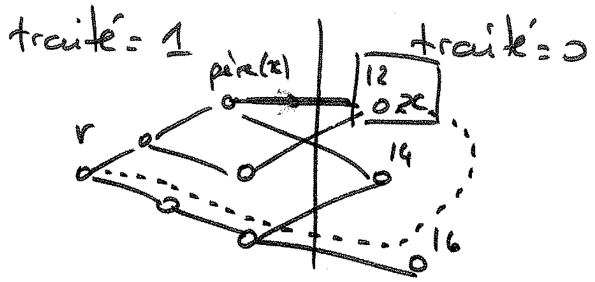
$$\left| \begin{array}{l} L_8 \to O(n) \\ L_{10} \text{ à } 13 \to O(d(x)) \leq O(n) \end{array} \right| n \text{ fois} \quad \underline{O(n^2)}.$$

Si on gère d par tas la L_8 prend $O(1)$, par contre chaque mise à jour de d prend $O(\log n)$.

le nombre total de boucle de la L_{10} est $\sum d(v) = 2m$

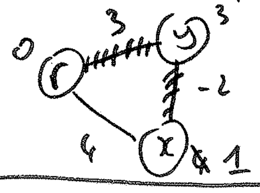
Donc $\underline{O(m \log n)}$.

- **Correction** : rapidement, par récurrence sur le nombre de sommets avec traité$(x) = 1$.



On regarde le x choisi L_8, si il y avait un p.c.c de r à x, il franchirait la "frontière" traité=1 / traité=0 et donc aurait une longueur ≥ au chemin choisi. ▨.

Rem :
- On obtient un arbre de p.c.c issu de r.
- Ça marche aussi en orienté ($Vois(x) \leftarrow Vois^+(x)$) : on obtient une arborescence sortante des p.c.c. issue de r.
- Si il y a des poids négatifs, Dijkstra ne marche pas



On trouve $d(r,y) = 3$ alors que rxy a longueur 2 !