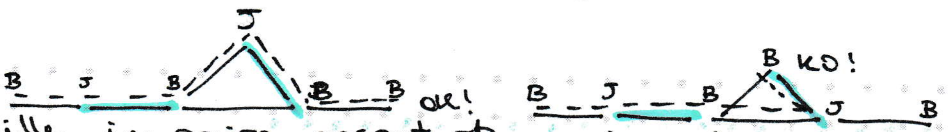
Cas pénible : 

→ cycles de taille impaire posent pb = selon le sens ils peuvent êtres B ou J.

On remplace le cycle de taille impaire par un sommet B.

On a un chemin améliorant → decompresser le bourgeon.

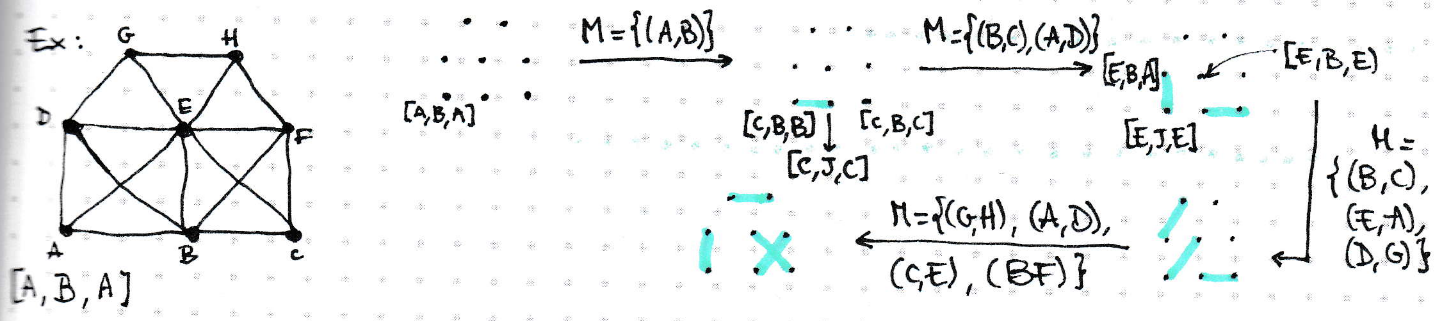
Algo d'Edmonds (recherche de chemin améliorant)

chaque sommet sera étiqueté par [r, c, p]
r : racine de l'arbre ∈V
c : couleur B/J
p : parent de l'arbre ∈V

C(|V|) • Retirer les étiquettes de ts les sommets qui en ont
C(|E|) • Marquer ttes les arêtes comme non explorées
C(|E|) • Repeter au choix
C(|E|)
- Ⓐ Trouver un sommet libre non etiqueté $v \in V$ et l'étiqueter $[v, B, v]$
- Ⓑ Trouver une arête (v, w) non explorée avec v d'étiquette $[r, B, p]$
  1) Marquer (v, w) comme explorée
  2) Si w n'a pas d'étiquette et est libre // chemin améliorant
     - P ← chemin de r à w
     - break
  3) Si w n'a pas d'étiquette et $\exists x$ tq $(w, x) \in M$ étiqueter w par $[r, J, v]$ et x par $[r, B, w]$
  4) Si w a l'étiquette $[s, B, q]$ avec $s \neq r$ // 2 arbre rejoints
     - P ← chemin de r à w + celui de w à s
     - break
  5) Si w a l'étiquette $[s, B, q]$ avec $s = r$ // cycle impair
     - remplacer les sommets du cycle par un seul sommet x dans le graphe et sauvegarder la correspondance sur une pile
     - étiqueter x par $[r, B, v]$
  6) Si w a pour étiquette $[s, J, p]$, ne rien faire.
- Si ni Ⓐ ni Ⓑ possibles, quitter la boucle avec $P \leftarrow \emptyset$

• Si $P \neq \emptyset$
C(|V|) dépiler la pile des bourgeons et remplacer les occurences d'état de substitut° par le chemin de taille paire approprié.

C(|E|)

Ex : 

Note : Ds un graphe biparti, il n'y a pas de cycle de taille impaire
→ l'algo d'Edmonds peut ê simplifié pour ne pas gerer les bourgeons.

Note : Un couplage maximum ds un graphe biparti peut aussi être vu comme un calcul de flot maximum.

Condit° necessaire au couplage parfait :

il existe un couplage parfait dans un graphe connexe si il y a un nbr pair de sommet

⚠ Ce n'est pas une condit° suffisante ! 

Thm de Tutte

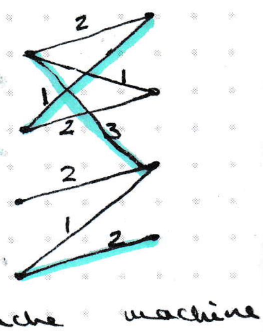
un graphe $G=(V,E)$ possède un couplage parfait ssi pour chaque sous ensemble $U \subseteq V$, le graphe induit $V \setminus U$ (on supprime les U de G) possède au plus $|U|$ composantes fortement connexes.

→ Ca ne donne pas un algo efficace (car il y a $2^{|V|}$ ss ensemble U)
Mais l'algo d'Edmonds donne un couplage maximum $O(|V|.|E|)$ et il suffit de regarder le nombre d'arêtes de M.

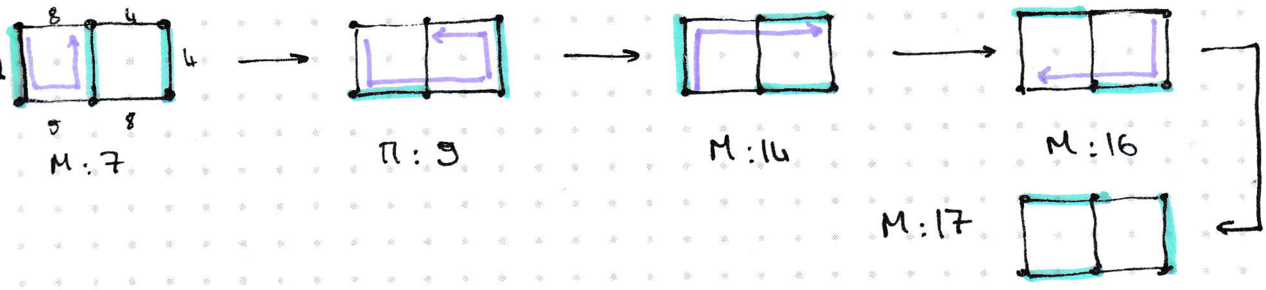
Couplage pondéré

Soit un graphe pondéré $G=(V,E,\omega)$ avec $\omega(e)$ le pds de $e \in E$
On cherche le couplage de pds maximum.

Chemin améliorant pondéré : chemin qui alterne arêtes de M et arêtes de $E \setminus M$. De plus, si le chemin commence ou termine sur une arête de $E \setminus M$, une extrémité de ces arêtes est libre. Enfin, la somme des poids des arêtes du chemin hors M dépasse celle des poids des arêtes du chemin M.



Ex :

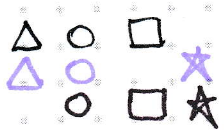


Algo d'Edmonds & Johnsons calcule un couplage pondéré maximum en $O(|V|^3)$. (aussi appelé algo hongrois)

Une couverture des sommets par les arêtes est un ss ensemble de E qui touche ts les sommets du graphe.

ex: arbre courant, couplage parfait, graphe lui m̂ (si sans sommet isolé)

Couverture minimale : on ne peut pas retirer d'arête
Couverture minimum : ∄ couverture avec moins d'arête

Ex : 

Chaque figure est declarée en pds couleurs. On veut emporter un nombre minimum d'objet représentant les formes et les couleurs

→ Couverture minimale

Algo pour construire une couverture minimale :

- calcul couplage maximum — $O(|V|.|E|)$
- pour chaque sommet — $O(|V|+|E|)$
  - ajoute arête sortante à celle du couplage
- ⇒ couverture minimale — $O(|V|.|E|)$

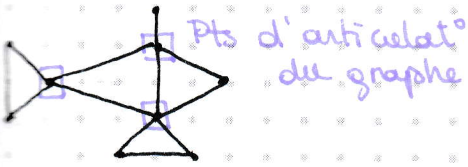
Problème du postier chinois (route inspect° pb)

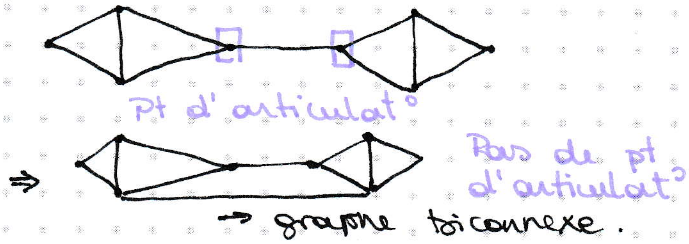
Dans un graphe G, on veut trouver le PCC qui :
- visite ttes les arêtes au - 1 x
- revient à son pt de départ

- Si un cycle eulérien existe, c'est une solut° du pb.
- Si il y a 2 sommets de degré impair ? Calcul PCC pour les relier. On double les arêtes le long du chemin → tous les d° sont pairs ⇒ cycle eulérien.
- Cas général : 2k sommet de d° impairs.

Algo de résolut° :
- $O(|E|+|V|)$ Soit S l'ensemble des sommets de d° impair
- $O(|V|^3)$ Construire Gs graphe complet de |S| sommets pondéré par les distances entre chaque sommet S ds le graphe d'origine
- $O(|V|^3)$ Calculer un couplage parfait de pds minimum pour GS
- $O(|E|)$ Doubler dans G les arêtes correspondant aux PCC select par couplage
- $O(|E|)$ Calculer circuit eulérien ds G

$O(|V|^3)$



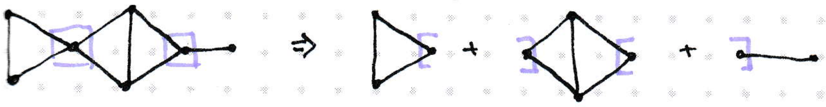


Def : Ds un graphe non orienté $G=(V,E)$, deux sommets $v_1 \in V$ et $v_2 \in V$ sont connectés s'il existe un chemin reliant v_1 à v_2

La relat° "est connecté" est une relat° d'équivalence
Ses classes d'équivalence (sommets interconnectés) sont appelées composantes connexes

Un sommet $v \in V$ est un pt d'articulat° (PA) si $G \setminus \{v\}$ a plus de composantes connexes que G

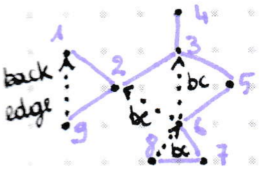
Un graphe est biconnexe s'il n'a pas de pt d'articulat°.
Sinon, il est séparable.



Cmt trouver les P.A. ds un graphe connexe?

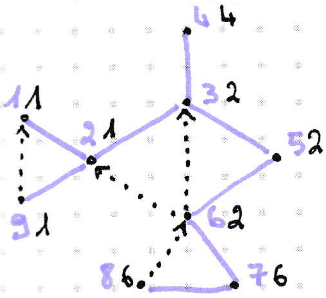
Idée 1 : Pour $v \in V$
si $G \setminus \{v\}$ n'est pas connexe (DFS)
alors v est un P.A

Pour DFS
$\Rightarrow O(|V| \times |E|)$



DFS construit arbre couvrant

Un noeud interne de l'arbre du DFS est un P.A de G
→ si $\exists$ un fils w de v dont le sous arbre ne contient aucun back edge remontant au dessus de v.



x Rank
x low

PA : 2, 3, 6

Racine : • si 1 fils $\Rightarrow$ pas P.A.
• Si ≥ 2 fils $\Rightarrow$ P.A.

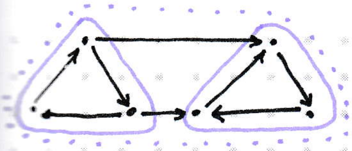
```
PA (G = (V, E))
    ∀ v ∈ V, rank [v] ← undef
    ∀ v ∈ V, low [v] ← undef
    ∀ v ∈ V, pred [v] ← undef
    index ← 1
    let v0 ∈ V                          // sommet de départ
    result ← ∅                          // ensemble de PA
    rec (v0)
    return result

def rec (v)
    children ← 0
    rank [v] ← low [v] ← index
    for w ∈ adj (v)
        if rank [w] is undef
            children ← children + 1
            pred [w] ← v
            rec[w]
            low[v] ← min (low[v], low[w])
            if (pred [v] is undef and children > 1)
                or ( pred[v] is not undef and low[w] ≥ rank[v])
                result . insert (v)
        else if w ≠ pred [v]
            low [v] ← min (low [v], rank [w])
```

$\Rightarrow \Theta(|V|+|E|)$

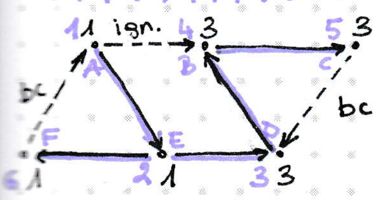
"Hopcroft - Tarjan"

Graphe orientés : Deux sommets $v_1 \in V$ et $v_2 \in V$ sont
- fortement connectés s'il $\exists$ un cycle orienté les reliant
- faiblement connectés si on peut les relier en ignorant l'orientation

 Classe d'eq : composantes fortement connexes (CFC, SCC) & composantes faiblement connexes (CfC, WCC)



2 SCC
1 WCC



Rank
low

Pile
A
E
~~F~~
~~B~~
~~C~~
~~D~~

{F, B, C}

{D, E, A}

Trajan :

```
entrée : graphe G = (V, E) orienté
sortie : partit° de V en SCC
∀ v ∈ V, rank [v] ← undef
∀ v ∈ V, low [v] ← undef
∀ v ∈ V, onstack [v] ← false       // pile DFS
index ← 1
S ← [ ]                            // pile sommets
Partit° ← ∅
for v ∈ V
    if rank [v] is undef
        rec (v)
return partit°
```

```
def rec (v)
    rank [v] ← low [v] ← index
    index ← index + 1
    onstack [v] ← true
    S.push (v)
    for w ∈ adj(v) :
        if rank [w] is undef
            rec (w)
            low [v] ← min (low [v], rank [w])
        else if onstack [w]
            low [v] ← min (low [v], rank [w])
    if rank [v] = low [v]                              //racine
        x ← ∅
        do
            w ← S.pop()
            x.insert (w)
        until (w = v)
        Partition.insert (x)
```

"2-SAT"

entrée : formule booléenne sous CNF avec clauses de taille 2
sortie : formule satisfiable ?

$(a \vee \bar{b}) \wedge (\bar{b} \vee c) \wedge (\bar{a} \vee \bar{b}) \wedge (c \vee d) \wedge (\bar{c} \vee e) \wedge (\bar{d} \vee \bar{e}) \wedge (\bar{c} \vee d)$

$a \vee b \equiv \bar{a} \Rightarrow b \equiv \bar{b} \Rightarrow a$

$a \bar{b} \bar{c} e \bar{d}$