Arbre non orienté connexe valué : L prim.

* Il conserve la connexité avec la source, proche de Djikstra.

KRUSKAN : Conserve l'acyclicité

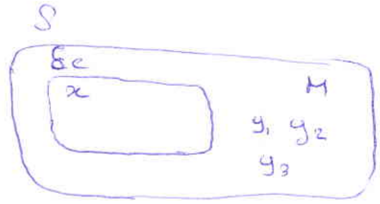
EDMONDS (Graphe orienté valué) : Input : Sommet de départ.
Output : Un arbre qui a pour racine l'Input.

02/04

PRIM : Soit G = <S, A, C> non-orienté connexe

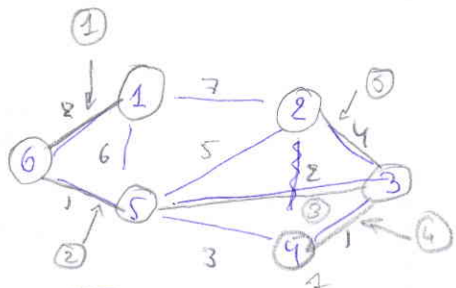
On travaille avec 2 ensembles :

- Sommets (S)
  - > Ec (
  - > M = S - Ec



algorithme procédure prim (entier x ; graphe g ; grapheT)

```
variables
    entier i, m, y, z        # m, y, z sont des sommets
    ensemble M
    reel r
    t_vectNent pp /* Plus proche sommet */
    t_vectNreel dist
Début
    /* Initialisation */
    T <- Graphevide
    M <- estun ensemblevide
    pour i <- 1 jusqu'à N faire :
        dist[i] <- coût(m, i, g)
        pp[i] <- x
        M <- ajouter(i, M)
    Fin pour
    M <- supprimer(x, M)
    /* Enrichissements successifs de T */
    Tant que M <> ensemblevide faire :
        m <- choisirmin(M, dist)
        M <- supprimer(m, M)
        z <- pp[m]
        r <- dist[m]
        T <- ajouterl'arete<m, z> de cout r à T
        pour i <- 1 jusqu'à d°(m, g) faire : *
            y <- i-ème-successeur-de m dans g
            si y ∈ M et cout(m, y, g) < dist[y] alors
                dist[y] <- cout(m, y, g)
                pp[y] <- m
            fin si
        fin pour
    fin tantque
Fin Algo Prim
```



	1	2	3	4	5	6
dist	∅	~~7~~ 5	~~∅~~ ~~4~~ 2	~~∅~~ 3	~~6~~	2
pp	1	~~1~~ 5	~~1~~ 5	~~1~~ 5	1	1

M = { ~~1~~, 2, 3, 4, 5, 6 }

	m	z	r	y *
①	6	1	2	1 ∉ M 5 ∈ M
②	5	6	1	1 ∉ M 2 ∈ M 3 ∈ M 4 ∈ M 6 ∉ M
③	3	5	2	2 ∈ M 4 ∈ M 5 ∉ M
④	4	3	1	3 ∉ M 5 ∉ M
⑤	2	3	4	1 ∉ M 3 ∉ M 5 ∉ M

	n°	dist	pp
update	2	4	3
update	4	1	3

Complexité :
Matrice : $\Theta(n)$
Liste : $\Theta(n^2)$
+ enrichissement : $\Theta(n)$