*Présentation rapide des fichiers et explications de la lecture du fichier et des 2 matrices*

Fichiers que nous avons rendus :

A2\_GrapheX : Les graphes que nous aurons à traiter mis à plat

A2\_TraceX : L’output de notre programme pour chaque graphe

Main.java : interface, on vous demande ce que vous souhaitez faire

Graphe.java : continent toutes les méthodes agissant avec les graphes

* On commence par récupérer le fichier, puis on initialise les variables nécessaires pour le programme
* On a un constructeur utilisé à la création de chaque graphe,
* On lit le fichier puis on l’affiche,
* On construit la matrice des valeurs et on l’affiche proprement,
* On convertit puis affiche la matrice adjacente,
* On détecte s’il y a un circuit ou non,
* On détermine les rangs avec un fonction qui affiche et qui retourne un tableau de sommets et son rang, en utilisant le même principe que pour la détection de circuit
* On affiche les entrées / sorties uniques,
* On vérifie si un ordonnancement est possible,
* Si l’ordonnancement est possible, on l’effectue et on affiche les résultats,
* Le tableau « date au plus tôt » va contenir : Autant de colonne qu'il y a de sommet dans le graphe, En première ligne le nom du sommet actuel, En 2nd ligne La liste des noms des prédécesseurs, En 3e ligne La liste des poids associés aux arcs provenant des prédécesseurs, En 4e ligne La liste des dates au plus tôt associés aux prédécesseurs, En 5e ligne La date au plus tôt associé au sommet actuel
* Création puis affichage du calendrier,
* Même méthode pour le tableau « date au plus tard ».

Sommet.java : Permet de parcourir les sommets au sein d’un graphe

* On commence par initialiser les variables utilisées
* Fonction permettant de parcourir le tableau des successeurs, on retourne un tableau vide.
* Getter pour récupérer les informations.
* Une fonction setter qui permettra de set le rang une fois que nous aurons vu que le calcul de rang est possible