GRAPHES : PLUS COURTS CHEMINS

1) Graphes non valués.

a) Créez un type file qui concrétise la structure abstraite de file, sachant que les seuls objets qu'on veut insérer dans la file sont les entiers $0, 1, \ldots, n-1$ (en un seul exemplaire maximum), n n'étant pas déterminé à l'avance, et sachant qu'on veut garantir une complexité constante dans les opérations classiques :
— new n crée une file pouvant contenir $0, 1, \ldots, n-1$
— add f x ajoute l'entier x à la file f
— take f retire l'élément le plus ancien de la file f et le retourne

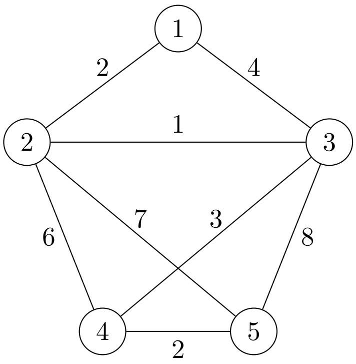
b) Créez un type ensemble avec ses opérations associées, qu'on veut en complexité constante :
— creer n crée un ensemble vide pouvant contenir les entiers $0, 1, \ldots, n-1$
— ajouter x e ajoute l'entier x à l'ensemble e en temps constant
— retirer e supprime dans l'ensemble e le plus ancien élément ajouté à e en temps constant
— appartenir x e calcule le booléen indiquant si l'ensemble e contient ou pas l'entier x en temps
— est_vide e calcule le booléen indiquant si l'ensemble e est vide ou pas en temps constant

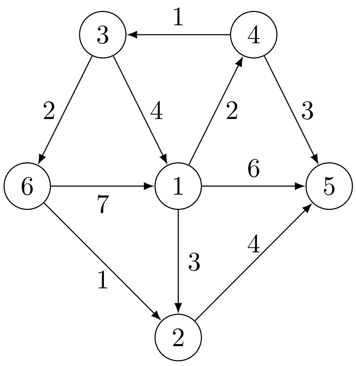
c) Un graphe non valué est donné par ses listes d'adjacences : écrivez une fonction distance x y g de paramètres x et y deux sommets et g un graphe, qui calcule la longueur du plus court chemin mesuré en nombre d'arcs/arêtes entre les deux sommets (avec la convention : -1 s'il n'existe aucun chemin de x à y)

d) Modifiez la fonction précédente pour qu'elle donne en plus un plus court chemin s'il en existe.

2) Floyd-Warshall : exemples.

Sur les exemples suivants, mettez en œuvre l'algorithme de Floyd-Warshall :





3) Floyd-Warshall

a) Écrivez la fonction de paramètre une matrice d'adjacence et qui met en œuvre l'algorithme de Floyd-Warshall.

b) Modifiez la fonction précédente pour qu'elle donne en plus un plus court chemin entre deux sommets.

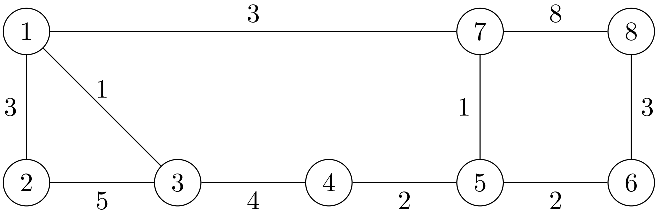
4) Point de stockage.

Un robot est chargé d'alimenter des machines à partir d'un point de stockage des matières premières : le robot ne peut alimenter qu'une machine à la fois, donc dès qu'il a alimenté une machine, il doit revenir au point de stockage pour reprendre une charge à livrer. Ce robot est astreint à se déplacer dans l'usine selon un plan précis.

Le plan de déplacement est représenté par un graphe et le point de stockage ainsi que les machines sont positionnées aux sommets du graphe. L'objectif est de placer le point de stockage en un point optimal : celui qui minimise la somme des distances entre le point de stockage et les machines.

Proposez un algorithme qui permet de résoudre le problème.

Sur l'exemple suivant, déterminez le point de stockage optimal :



Écrivez une fonction CAML de paramètre le graphe (donné par une matrice d'adjacence) qui résout le problème.