**Algorithme du point fixe**

algorithme

fonction pointFixe(Graphe g, Nœud depart) : Valeur   
début  
 valeurs.setValeur(depart) 🡨 0  
 pour n de 1 à g.listNoeud().size() faire  
 valeurs[n].distance 🡨 +∞  
 valeurs[n].parent 🡨 Indéfini  
 finpour

Q 🡨 g.listNoeud()

changement 🡨 vrai  
 tant que changement faire  
 changement 🡨 faux   
 pour i de 0 à Q.size()-1 faire  
 pour chaque Arc a de Q.get(i) faire  
 double newValeur 🡨 valeurs.getValeur(Q.get(i)) + a.getCout()  
 si valeur < valeurs.getValeur(a.getDest()) alors  
 valeurs.setValeur(a.getDest(), newValeur)  
 valeurs.setParent(a.getDest(), Q.get(i))  
 changement 🡨 vrai  
 fsi  
 finpour  
 finpour  
 fintantque

retourne valeurs

fin

**Lexique :**

g : Gaphe, le graphe dont on veut connaître les chemins minimaux depuis un nœud de départ

depart : Nœud, le nœud de départ depuis lequel on veut connaître les chemins minimaux dans le graphe

valeurs : Valeur, les valeurs des chemins minimaux

n : entier, indice d’itération de la liste de Nœud du graphe

Q : Liste(Nœud), la liste de nœuds du graphe g

changement : booléen, si la variable valeurs change (donc que l’on n’a pas atteint le point fixe) alors vrai, sinon faux

i : entier, indice d’itération qui permet le parcours des nœuds du graphe

a : Arc, parcours de chaque arc d’un nœud

newValeur : double, correspond à la valeur de l’arc courant + le cout du chemin le plus court actuel