TD parcours de graphes
Algorithme A star
ECL - sept. 2022

(Version élèves)

ASG

1 Un algorithme

1.1 Algorithme de principe

L'algorithme suivant est plus complet pour la stratégie de parcours **A_star**.
☞ Rappel : si la valeur h(n) = 0, on sera en présence d'un algorithme de Dijkstra.

La structure de chaque noeud du graphe.

classe **Noeud** contiendra :

(x, y) : : les coordonnées du noeud dans un espace 2D
degre : le degré (nb connexions) du noeud
. . . Autres informations éventuelles concernant le noeud
liste_noeuds_parents : liste de noeuds (leur x,y) depuis la racine jsq ce noeud
g : valeur de g
h : valeur de h
➛ **f** se déduit de **g** et **h**

L'algorithme du proncipe de Astar peut être :

```
Fonction parcours_A_star :
Entrée : G : Graphe, Départ, Destination : Noeud
Sortie : Trajet : Chemin (= vide si pas de solution)
Debut
   Pour tout noeud N : N.g=infini
   Pour tout noeud N : N.f=infini
   Pour tout noeud N : N.h=distance(N → Destination)     # euclidienne par défaut
   Départ.g=0
   Départ.f=Départ.h     # distance euclidienne par défaut
   Coming_From={}     # pour le Trajet
   déjà_visités={}     # ensemble vide noté "set()" en Python
   File_attente=[Départ]     # Une liste contenant "Départ"
   Tant que File_attente != [] :
      trier File_attente selon la valeur de f des noeuds
      noeud = tete_de(File_attente)     # la tête de la file est retirée
      Si noeud = Destination     # identité des points (x,y) et (x',y')
      Alors renvoyer construire_trajet(noeud, Coming_From)     # non fourni
      liste_des_voisins=voisins(noeud, G)
      Si liste_des_voisins = []
      Alors passer à l'itération suivante     # "continue" en Python
      Pour tout V dans liste_des_voisins :
         estimation_valeur_g_de_V= noeud.g + distance(noeud → V)     # euclidienne
         Si estimation_valeur_g_de_V < V.g
         Alors Coming_From[V] = noeud
               V.g = estimation_valeur_g_de_V
               V.f=V.g+V.h
               enfiler(V)
               ajouter V à déjà_visités
      renvoyer []     # ECHEC
   Fin Tq
Fin parcours_A_star
```