Algorith utilisé: A\*

- À chaque coordonnée x du labyrinthe, on definie une valeur heuristique h(x) qui est la distance à vol d'oiseau entre cette coordonnée x et la coordonnée finale x\_end.

- Au debut, on enregistre à la position de départ, le poids est h(x\_start); tous les autres coordonnées sont infinites.

- À chaque fois, on:

+ Choisit des noeuds x qui a le poids cumulé f(x) minimal (au debut, la liste contient seulement noeud de départ)

+ Calcule le poid nouveau des noeuds adjacents y de x (dans notre case, c'est les 8 coordonnées autours de x):

f(y) = p(y) + h(y) avec p(x) est la distance "plus court" on a trouvé pour aller à y

+ Si ces poids sont inférieurs à ses valeurs précédents, on les enregiste et les ajoute dans la file de priorité

-> Notre boucle est terminée si la position x concide avec le position final ou la file de priorité est vide.

