

$D^* - A^* - Dijkstra \leftarrow \text{Graph-based} \leftarrow \text{path-planning}$

$BIT^* - \text{informed RRT} - RRT^* - RRT \leftarrow \text{Sampling-based}$

در مسأله جابجایی باید ابتدا گراف را مشخص کنیم:

// pseudo code

if (front == 0) { move forward (1) }

While

if (front != 0) {

nodes.add (Read (GPSSensor)) → if (node exists) { return 0; }

while (front != 0) { turnRight (1) }

برای پیدا کردن در ب خروج، در صورتی که اگر به nodes خالی نباشد و از آخرین نقطه ای که سنسور Left+ برده 0 بوده حد اکثر 90cm دیگر 0 باشد، در ب است و با اضافه کردن مقداری به سمت چپ نقطه آن را به عنوان خود end به nodes اضافه میکنیم. در غیر اینصورت باید به آخرین نقطه ای که Left+ برده برگردیم و turnLeft را اجرا کنیم.

با کامل شدن گراف (داشتن نقطه تکراری) در نقطه شروع هستیم.

برای اضافه کردن نقاط و طاق، به فاصله بین نقاط گوشه اتاق، نقاط دیگری را به nodes اضافه میکنیم (Grid) پس با استفاده از الگوریتم Dijkstra مسیر را پیدا کرده و نقاط بازدهش را از nodes حذف میکنیم تا به نقطه end برسیم. * نقاط end را در حافظه ذخیره میکنیم که اتاق تکراری انتخاب نشود.

// pseudo code

if (f == 0 && R == 1) { move forward (1) }

else { turnRight (1) } → 90deg

Random Mouse

← Right Hand / Left Hand

BFS - DFS

A^*

Dijkstra

← گوییم حای حل
Maze