

پروژه: حل مسئله مسیریابی با استفاده از الگوریتم‌های جستجوی آگاهانه

هدف پروژه:

هدف این پروژه، استفاده از دو الگوریتم جستجوی آگاهانه A^* و IDA^* برای حل یک مسئله مسیریابی است. شما باید از یک نقطه شروع به یک نقطه هدف در یک گراف حرکت کنید و کمترین هزینه را برای رسیدن به مقصد پیدا کنید. این پروژه به شما کمک می‌کند تا مفاهیم جستجوی آگاهانه، هیوریستیک‌ها، و جستجوی عمق تدریجی را عملی تجربه کنید.

شرح مسئله:

شما در یک شبکه شهری از تقاطع‌ها و خیابان‌ها قرار دارید که باید از نقطه‌ای به نقطه‌ای دیگر حرکت کنید. هر تقاطع به عنوان یک گره در نظر گرفته می‌شود و هر خیابان به عنوان یک یال با هزینه مشخص (مثلاً مسافت یا زمان) مدل‌سازی می‌شود. هدف شما این است که کمترین هزینه طی کردن مسیر از نقطه شروع به نقطه هدف را پیدا کنید.

در این پروژه، شما دو الگوریتم جستجوی آگاهانه A^* و IDA^* را پیاده‌سازی خواهید کرد تا کمترین هزینه را برای حرکت از نقطه شروع به نقطه هدف پیدا کنید.

ورودی پروژه:

- **گراف (شهر):** گراف به صورت یک دیکشنری از تقاطع‌ها (گره‌ها) و خیابان‌ها (یال‌ها) با هزینه‌ها (مسافت یا زمان) نمایان می‌شود. این گراف قابل تغییر باشد و قابلیت خواندن از فایل وجود داشته باشد.
- **نقطه شروع و هدف:** دو گره که مسیر از آن‌ها باید پیدا شود.

گراف (شهر با تقاطع‌ها و خیابان‌ها):

در این پروژه، گراف به عنوان یک شبکه از تقاطع‌ها (گره‌ها) و خیابان‌ها (یال‌ها) مدل‌سازی می‌شود. در واقع، گراف نشان‌دهنده‌ی شبکه‌ای از جاده‌ها یا مسیرها است که می‌خواهیم در آن حرکت کنیم.

توضیح گراف:

1. **تقاطع‌ها (گره‌ها):** هر تقاطع در شهر را به عنوان یک گره در نظر می‌گیریم. تقاطع‌ها می‌توانند در هر نقطه‌ای از نقشه شهر قرار بگیرند. به عنوان مثال، یک تقاطع می‌تواند موقعیت $(0, 0)$ یا $(2, 1)$ داشته باشد که به ترتیب نشان‌دهنده‌ی مختصات این تقاطع در مختصات 2D است.
2. **خیابان‌ها (یال‌ها):** هر خیابان که بین دو تقاطع قرار دارد، به عنوان یک یال در گراف در نظر گرفته می‌شود. یال‌ها معمولاً هزینه دارند که می‌تواند به شکل مسافت یا زمان حرکت در نظر گرفته شود. برای هر خیابان می‌توان هزینه‌ای برای طی کردن آن تعیین کرد. به عنوان مثال، هزینه حرکت از تقاطع $(0, 0)$ به $(0, 1)$ می‌تواند برابر با 1 باشد (که این ممکن است مسافت یا زمان حرکت باشد).

نمونه گراف:

فرض کنید در یک شهر شش تقاطع داریم که موقعیت‌هایشان به صورت مختصات دو بعدی (x, y) نمایش داده شده است. گراف اینگونه خواهد بود:

```
graph = {  
  (0, 0): [((0, 1), 1), ((1, 0), 1.5)],  
  (0, 1): [((0, 0), 1), ((1, 1), 1)],  
  (1, 0): [((0, 0), 1.5), ((1, 1), 1), ((2, 0), 2)],  
  (1, 1): [((0, 1), 1), ((1, 0), 1), ((2, 1), 1.5)],  
  (2, 0): [((1, 0), 2), ((2, 1), 1)],  
  (2, 1): [((1, 1), 1.5), ((2, 0), 1)]  
}
```

خروجی پروژه:

- مسیر بهینه: مسیر بهینه از نقطه شروع به هدف که کمترین هزینه را دارد.
- هزینه کل: هزینه کل مسیر بهینه.
- گزارش گره‌های بازدید شده: تعداد گره‌هایی که در هر الگوریتم جستجو بازدید شده‌اند.

نکات تکمیلی:

- پروژه در گروه‌های نهایتاً دو نفره می‌تواند پیاده‌سازی شود.
- در انتخاب زبان برنامه‌نویسی محدودیتی وجود ندارد.