

مستندات بخش پیاده‌سازی

علی علی‌محمدی

۹۶۱۳۰۲۷

- در فایل فشرده‌ی zip، دو فایل py به زبان Python وجود دارد؛ UCS.py مربوط به الگوریتم جست‌وجوی هزینه‌ی یکنواخت است و A-Star.py مربوط به الگوریتم A^* است.
- تنها کتاب‌خانه‌ی مورد نیاز برای اجرای برنامه‌ی الگوریتم A^* ، کتاب‌خانه‌ی NetworkX 2.5 است و برنامه‌ی جست‌وجوی هزینه‌ی یکنواخت، پیش‌نیازی ندارد.
- از پایتون نسخه‌ی 3.8 استفاده شده است.
- برای اجرای برنامه، لازم است فایل Environment.txt در همان پوشه‌ای قرار گیرد که فایل‌های پایتون در آن قرار دارند.
- در هر برنامه، ابتدا فایل ورودی خوانده می‌شود و پس از آن، هرکدام از خانه‌های ورودی را به عنوان یک رأس در گراف در نظر می‌گیریم. سپس با توجه به مقادیر خانه‌ها، یال‌های جهت‌دار لازم را بین رئوس قرار می‌دهیم. در برنامه‌ی UCS، پیاده‌سازی گراف را خودمان انجام می‌دهیم اما در برنامه‌ی A^* ، برای سهولت کار، از کتاب‌خانه‌ی NetworkX استفاده می‌کنیم.
- در برنامه‌ی A^* ، هیورستیک پیشنهادی ما، هیورستیک فاصله‌ی منهتن $h = \Delta x + \Delta y$ است؛ زیرا حرکت‌های قطری مجاز نیستند و فقط قادر به حرکت عمودی یا افقی هستیم.
- کامنت‌های برنامه به وضوح روند کار را تشریح می‌کنند. خروجی هر دو تا برنامه لزوماً یکسان نیست اما حداقل هزینه در آن‌ها برابر است. همچنین در حالی که زمان اجرای برنامه بسیار ناچیز است، امکان مقایسه‌ی دقیق روی سیستم را نداریم اما به نظر می‌رسد که در این دو مثال خاص، الگوریتم A^* ، اندکی سریع‌تر از الگوریتم UCS بود.
- در پایان هر دو بخش (الف) و (ب) مذکور در صورت سؤال، مسیر بهینه، حداقل هزینه و مدت زمان اجرای آن بخش از برنامه نمایش داده می‌شود. دو نمونه از خروجی‌ها در زیر آورده شده است:

قسمت (ب) برنامه‌ی A^*

----- Part B -----

(17,1)

(18,1)

.

.

.

(16,29)

(17,29)

Minimum Cost: 78

Execution Time: 15.62643051147461 Milliseconds

قسمت (الف) برنامه‌ی UCS

----- Part A -----

(0,0)

(1,0)

.

.

.

(22,24)

(23,24)

Minimum Cost: 93

Execution Time: 0.0 Milliseconds

در صورت وجود هرگونه سؤال یا ابهام، با ایمیل alialimohammadi@aut.ac.ir در تماس باشید.

¹ Manhattan Distance Heuristic