

تمرین هوش مصنوعی

مریم نادری - ۹۵۲۰۲۳۰۳۵

الگوریتم های جستجوی a Star و hill climbing و simulated annealing در فایل Search.py پیاده سازی شده اند که از آن ها در eightpuzzle.py استفاده می کنیم. قسمت های مختلف کد توسط کامنت های متعدد توضیح داده شده است.

با اجرا کردن فایل eightpuzzle.py، دوازده پازل-۸ به صورت رندوم generate میشود و نمایش داده میشود و path پیدا شده توسط الگوریتم های جستجوی مختلف، به همراه هزینه ی بهینه زیر هر پازل در خروجی نمایش داده میشود.

قسمت الف-

برای ۱۲ نمونه تصادفی از پازل های رندوم، درصد رسیدن به پاسخ توسط الگوریتم های مختلف محاسبه شده:

A star: 100%

Hill climbing: 0%

Simulated annealing: 8%

همانطور که از این درصد ها، و همچنین با اجرا کردن برنامه نیز به وضوح میتوان دید، الگوریتم های مختلف تپه نوردی به ندرت به goal میرسند و معمولاً در یک اکستریم محلی گیر می افتند. الگوریتم simulated annealing نسبت به hill climbing پیشروی بیشتری می کند ولی باز هم با درصد کمی به goal میرسد.

قسمت ب-

برای ۱۲ نمونه تصادفی از پازل های رندوم، تعداد کل action هایی که تا پایان یافتن الگوریتم، اتخاذ می شود، به ازای الگوریتم های مختلف برابر است با:

A star: 89

Hill climbing: 20

Simulated annealing: 154

به ازای هر action یک حالت از حالت های مختلف پازل-۸ حاصل میشود. تعداد کل حالت های مختلف برابر جایگشت ۹ خانه ی این پازل (با احتساب جایگاه خانه ی خالی) است که برابر است با ۳۶۲۸۸۰

بدیهی است که قسمت در دسترس فضای حالت در مقایسه با کل فضای حالت، برای هر سه الگوریتم بسیار ناچیز است.

#### قسمت ج-

در این بازی وقتی دو نفر به صورت رقیب در مقابل هم قرار میگیرند، مانند الگوریتم minimax تنها در صورتی به جواب میرسیم که فرض کنیم رقیب نیز تمام سعی خود را می کند که امتیاز خود را ماکسیمم و امتیاز رقیب را مینیمم کند.