

به نام خدا



**دانشگاه صنعتی امیرکبیر**  
(پلی تکنیک تهران)

گزارش پروژه ی اول درس هوش مصنوعی

نام کامل نویسنده:

مرتضی دامغانی نوری

(۹۶۲۵۸۰۱)

نام استاد درس:

استاد روشن فکر

اردیبهشت ماه هزار و چهار صد

## نحوه ی مدل سازی مساله برای جستوجو:

نحوه ی مدل سازی به این شکل است که ابتدا الگوریتمی که کاربر قصد جستوجوی ورودی مساله با آن را دارد از کاربر گرفته می شود. سپس با الگوریتم مورد نظر، جستوجوی ماتریس ورودی از محلی که ربات در آن قرار گرفته است، شروع می شود تا زمانی که کره ای توسط ربات پیدا شود، ادامه میابد. بعد از آن که کره پیدا شد، ربات در پشت آن قرار می گیرد و این بار به کمک الگوریتم مورد نظر خانه ی goal جستوجو می شود و در هر مرحله ربات با توجه به خانه ای که توسط الگوریتم مورد نظر پیشنهاد می شود، کره را در صورت امکان به آن خانه هل می دهد و این کار تا زمانی ادامه پیدا می کند که کره به خانه ی Goal برسد.

## تابع شهودی انتخاب شده و بررسی قابل قبول بودن آن:

تابع شهودی ای که برای پیاده سازی الگوریتم  $A^*$  مورد استفاده قرار گرفته است، تابع فاصله ی منهتن است که با توجه به اثبات هایی که در کلاس انجام شده است، این تابع قابل قبول (admissible) است.

## توضیح کلی توابع و کلاس های تعریف شده در کد:

کلاس هایی که در کد مورد استفاده قرار گرفته اند:

## کلاس Node:

این کلاس توصیف کننده ی گره هایی است که در گراف جستوجو تولید می شوند.

## کلاس Location:

این کلاس توصیف کننده ی مکان ها در ماتریس ورودی است.

توابعی که در کد مورد استفاده قرار گرفته اند:

## تابع ids\_algorithm:

تابع اصلی ای است که به کمک آن الگوریتم IDS پیاده سازی می شود.

## تابع ids\_path\_finder:

تابعی است که یک نقطه ی شروع و یک نقطه ی هدف را به عنوان ورودی می گیرد و مسیری را از نقطه ی شروع تا نقطه ی هدف بر اساس الگوریتم IDS به عنوان خروجی باز می گرداند. در صورت عدم موفقیت در پیدا کردن چنین مسیری نیز failure باز می گرداند.

### تابع `bidirectional_bfs_algorithm`:

تابع اصلی ای است که به کمک آن الگوریتم Bidirectional BFS پیاده سازی می شود.

### تابع `bidirectional_bfs_path_finder`:

تابعی است که یک نقطه ی شروع و یک نقطه ی هدف را به عنوان ورودی می گیرد و مسیری را از نقطه ی شروع تا نقطه ی هدف بر اساس الگوریتم Bidirectional BFS به عنوان خروجی باز می گرداند. در صورت عدم موفقیت در پیدا کردن چنین مسیری نیز failure باز می گرداند.

### تابع `a_star_algorithm`:

تابع اصلی ای است که به کمک آن الگوریتم  $A^*$  پیاده سازی می شود.

### تابع `a_star_path_finder`:

تابعی است که یک نقطه ی شروع و یک نقطه ی هدف را به عنوان ورودی می گیرد و مسیری را از نقطه ی شروع تا نقطه ی هدف بر اساس الگوریتم  $A^*$  به عنوان خروجی باز می گرداند. در صورت عدم موفقیت در پیدا کردن چنین مسیری نیز failure باز می گرداند.

## تابع output\_file\_generator:

تابعی است که از آن برای تولید فایل های خروجی با فرمت گفته شده و اطلاعات لازم استفاده می شود.

## مقایسه ی روش های پیاده سازی شده:

	Time (s)	Time Complexity	Generated Nodes	Opened Nodes	Butter Finding Depth	Goal Finding Depth	Cost
<b>IDS</b>	0.97	$O(b^m)$	567	246	4	51	635
<b>Bidirectional BFS</b>	0.15	$O(b^{d/2})$	176	44	8	36	22
<b>A*</b>	0.13	$O(b^d)$	134	43	5	38	54

اطلاعات مربوط به اجرا الگوریتم ها برای ورودی test3.txt که دارای دو کره برای رسانده شدن به هدف بود در جدول بالا آورده شده است.