



مرحله اول پروژه

مرحله اول پروژه درس از دو بخش تشکیل شده است. در بخش اول باید از الگوریتم‌های جستجو برای پیاده‌سازی یک بازیکن هوشمند در بازی معروف Pacman استفاده کنید و در بخش دوم با پیاده‌سازی الگوریتم کاهش گرادیان یک مدل رگرسیون خطی برای مسأله داده شده آموزش داده و آن را مورد ارزیابی قرار دهید. توجه کنید که باید برای پروژه خود بعد از پیاده‌سازی گزارش تهیه کرده و در آن پیاده‌سازی خود را توضیح داده و نتایج را نمایش دهید. فایل‌های پروژه در این [لینک](#) قرار دارند.

۱- بازی Pacman

در این بخش از پروژه می‌خواهیم عامل هوشمندی پیاده‌سازی کنیم که در مورد نحوه دستیابی به غذاهای موجود در بازی معروف Pacman تصمیم‌گیری کند. در این مسئله که Corner Problem نام دارد، از الگوریتم‌های جستجوی آگاهانه و ناآگاهانه برای هدایت عامل Pacman به سمت غذاها استفاده می‌شود. الگوریتم‌های در نظر گرفته شده برای پیاده‌سازی UCS، DFS و A^* هستند. شرط برنده شدن این است که عامل Pacman همه غذاهای موجود در محیط را مصرف کند.

۱-۱- مراحل پیاده‌سازی

برای تسهیل پیاده‌سازی عامل، ساختار کد عامل از قبل ایجاد شده و شما باید فقط بخش‌های مرتبط با الگوریتم‌های جستجو را که در زیر مشخص شده است تکمیل کنید.

۱. به فایل `search.py` مراجعه کنید. در این فایل سه تابع وجود دارد که شما باید پیاده‌سازی آن را تکمیل کنید:

- تابع `depthFirstSearch` محل پیاده‌سازی الگوریتم DFS
- تابع `uniformCostSearch` محل پیاده‌سازی الگوریتم UCS
- تابع `aStarSearch` محل پیاده‌سازی الگوریتم A^*

۲. تابع اکتشافی مورد استفاده در الگوریتم A^* در تابع `cornersHeuristic` در ابتدای فایل `searchAgents.py` تعریف می‌شود که باید آن را پیاده‌سازی کنید. در پیاده‌سازی این تابع می‌توانید از توابع تعامل با محیط که در اختیار شما قرار گرفته است استفاده کنید.

توجه کنید که توابع پیاده‌سازی شده باید بتوانند در نمونه‌های مختلفی از مسأله Corner Problem بکار گرفته شوند. نمونه‌هایی که می‌توانند از نظر ابعاد محیط، تعداد غذا و موقعیت اولیه Pacman متفاوت باشند. به همین دلیل سه محیط اولیه در اختیار شما قرار داده شده است تا برای آزمایش توابع پیاده‌سازی شده مورد استفاده قرار گیرد و در ادامه توضیح داده شده است.

۱-۲- محیط‌ها

- محیط Simple Corner: این محیط، ساده‌ترین محیط است و می‌توانید برای ارزیابی صحت پیاده‌سازی الگوریتم‌ها از آن استفاده کنید. جهت اجرای این محیط با هر یک از الگوریتم‌های جستجوی پیاده‌سازی شده می‌توانید از دستورات زیر استفاده کنید:

```
python pacman.py -l simpleCorner -p SearchAgent -a
fn=dfs,prob=CornersProblem

python pacman.py -l simpleCorner -p SearchAgent -a
fn=ucs,prob=CornersProblem

python pacman.py -l simpleCorner -p SearchAgent -a
fn=astar,prob=CornersProblem,heuristic=cornersHeuristic
```

- Hard Corner: این محیط نسبت به محیط simple corner پیچیده‌تر است و می‌تواند برای ارزیابی تابع اکتشافی مورد استفاده در الگوریتم A^* به صورت زیر مورد استفاده قرار گیرد:

```
python pacman.py -l hardCorner -p SearchAgent -a
fn=astar,prob=CornersProblem,heuristic=cornersHeuristic
```

- Big Corner: این محیط دارای ابعاد بزرگتری نسبت به محیط‌های قبلی بوده و به صورت زیر قابل فراخوانی است:

```
python pacman.py -l BigCorner -p SearchAgent -a
fn=astar,prob=CornersProblem,heuristic=cornersHeuristic
```

۱-۳- نحوه ارزیابی

هر بار پس از اجرای محیط، یک گزارش در ترمینال چاپ می‌شود که نشان‌دهنده امتیاز کسب‌شده، زمان اجرا و تعداد گره‌های بسط داده شده است. ارزیابی کیفیت تابع اکتشافی پیاده‌سازی شده براساس تعداد گره‌های بسط داده شده توسط الگوریتم جستجوی A^* خواهد بود، که در جدول زیر امتیاز مرتبط برای محیط‌های Hard Corner و Big Corner نشان داده شده است:

Score	Expanded Nodes in BigCorner	Expanded Nodes in HardCorner
100%	Less than 2000	Less than 800
80%	Less than 3000	Less than 1200
60%	Less than 4500	Less than 1500

همانطور که قبلاً گفته شد در ارزیابی پیاده‌سازی انجام شده از نمونه محیط‌های دیگری نیز استفاده می‌شود که انتظار می‌رود تعداد گره‌های بسط داده شده در حد قابل قبول باشد. بنابراین علاوه بر صحت پیاده‌سازی الگوریتم‌های DFS، UCS و A^* در فایل search.py، تابع اکتشافی پیاده‌سازی شده در searchAgents.py نیز از نظر تعداد گره‌های بسط داده شده مورد بررسی قرار خواهد گرفت.