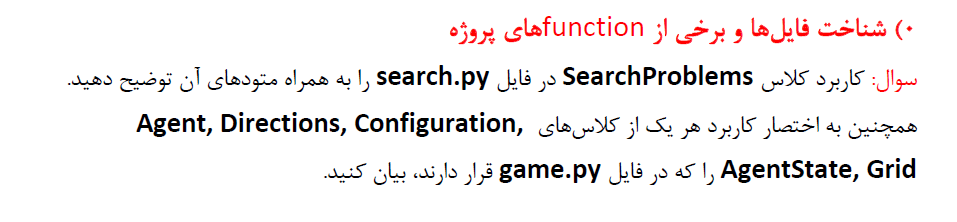
"بسمه تعالی"

مبانی هوش کاربردی – پروژه اول



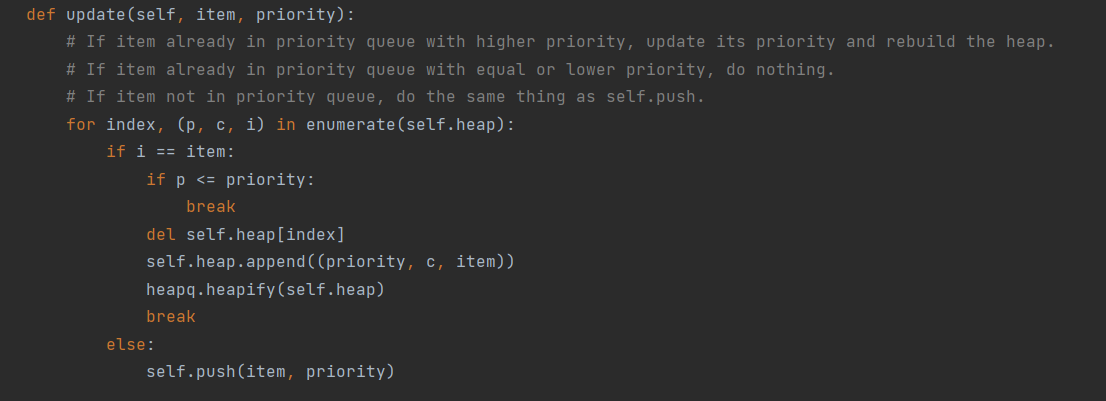
این کلاس بصورت کلی ساختار یک مسئله جستجو را مشخص میکند

* getStartState
* حالت اولیه را برای مسئله جستجو برمیگرداند
* isGoalState
* درصورتی که در حالت قابل قبول هدف باشیم مقدار صحیح را برمیگرداند
* getSuccessors
* یک سه گانه را برمیگرداند که شامل "وارث" یا بعبارت بهتر حالت ممکن بعدی، هزینه رسیدن به این حالت و عملی که باید انجام شود تا به آن برسیم خواهد بود
* getCostOfActions
* براورد هزینه کلی برای مجموعه ای از اعمال را برمیگرداند
* tinyMazeSearch
* دنباله ای از حرکات را که یک ماز کوچک را حل میکنند برمیگرداند، برای باقی ماز ها جواب درستی نخواهد داشت.
* depthFirstSearch
* با الگوریتم DFS عمیق ترین نود هارا در درخت جستجو میکند
* breadthFirstSearch
* با الگوریتم BFS سطح به سطح جستجو میکند
* uniformCostSearch
* با الگوریتم UCS نود ها با کمترین هزینه را جستجو میکند
* nullHeuristic
* heuristic را محاسبه میکند ( هزینه به نزدیک ترین حالت هدف ممکن)
* aStarSearch
* نود ای را جستجو میکند که بصورت برایند کمترین هزینه و اولین heuristic را دارد

**کلاس ها در game.py**

* Agent
* کلاس اجنت متدی تحت عنوان getAction دارد که با دریافت حالت فعلی و عملی برای آن وارد عمل میشود
* Configuration
* موقعیت فعلی را حفظ میکند در عین اینکه حرکت را دنبال میکند
* AgentState
* حالت فعلی یک اجنت را نگه داری میکند
* Directions
* موقعیت های حرکتی را تعریف میکند
* Grid
* موقعیت خانه ها در بازی

0)



در این بخش در یک حلقه روی آیتم های موجود در هیپ حرکت میکنیم، اگر آیتم i در هیپ موجود بود و اولویت کمتر یا مساوی داشت کاری نمیکنیم ولی اگر غیر این بود آن را از هیپ پاک کرده و مجددا هیپ را میسازیم و اگر این آیتم جدید بود آن را پوش میکنیم