

به نام خدا



درس هوش مصنوعی و سیستم های خبره

تمرین برنامه نویسی دوم

مدرس درس:

جناب آقای دکتر محمدی

طراحان:

سایین اعلا - عرفان زارع

مهلت ارسال: ۱۴۰۲/۰۸/۱۲

مقدمه

سودوکو یک جدول ۹×۹ است که کل جدول هم به ۹ جدول کوچک تر ۳×۳ تقسیم شده است. در این جدول چند عدد به طور پیش فرض قرار داده شده که باید باقی اعداد را با رعایت سه قانون زیر یافت:

قانون اول: در هر سطر جدول اعداد ۱ الی ۹ بدون تکرار قرار گیرد.

قانون دوم: در هر ستون جدول اعداد ۱ الی ۹ بدون تکرار قرار گیرد.

قانون سوم: در هر ناحیه ۳×۳ جدول اعداد ۱ الی ۹ بدون تکرار قرار گیرد.

5	3			7				
6			1	9	5			
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

پروژه سری شماره ۲ پیرامون مبحث csp و backtracking می باشد. در این پروژه هدف این است که جدول سودوکو را تولید و آن را با روش های خواسته شده حل نمایید.

مراحل پیاده سازی

۱. ابتدا cell مربوط به libraries را ران نموده تا کتابخانه های مورد نیاز import شوند. در صورت عدم نصب داشتن هر کدام از کتابخانه ها می توانید با دستور pip install و نوشتن نام کتابخانه مدنظر، آن را نصب نمایید. (۲نمره)
۲. حال در تابع generate_sudoku باید یک جدول سودوکو ایجاد کنید، با توابع نوشته شده، آنرا حل نموده و در نهایت تعدادی از عناصر را حذف و جدول را خروجی دهید. ورودی این جدول تعداد سطر و ستون جدول مدنظر است. (۵نمره)
۳. در تابع fill_box، به پر کردن جدول با اعداد رندوم بپردازید. (۸نمره)
۴. در دو تابع کامل شده fill_diagonal و is_safe به ترتیب به پر کردن جدول به سختار نوشته شده و بررسی درست و قابل حل بودن جدول تولیدی می پردازیم.
۵. در تابع used_in_row بررسی می شود که آیا در سطر بیان شده، عدد مدنظر موجود هست یا خیر. (۷ نمره)
۶. در تابع used_in_column بررسی می شود که آیا در ستون بیان شده، عدد مدنظر موجود هست یا خیر. (۷ نمره)
۷. در تابع used_in_box بررسی می شود که آیا در بسته بیان شده، عدد مدنظر موجود هست یا خیر. (۷ نمره)

۸. در تابع `find_unassigned_location` بررسی می شود و خانه های تعیین نشده در جدول را مشخص می کند. در صورتی که مقدار اولیه تعیین شده بود مقدار ۱-و ۱- برگردانید. (۴نمره)
۹. در تابع `remove_elements` عناصر رو از جدول حذف می کنند. (۱۰ نمره)
۱۰. در تابع `solve_sudoku` با توجه به توابع نوشته شده و کامل شده به تابع حل جدول ورودی میپردازیم. در این تابع با استفاده از روش `backtracking` به حل جدول می پردازد. (۲۰ نمره)
۱۱. در تابع `display_grid` به نمایش جدول می پردازیم. (۵نمره)
۱۲. در `solve_sudoku_csp` به حل جدول با `csp` می پردازیم.
۱۳. در تابع `create_domains` ما به ایجاد دامنه ای برای هر سلول جدول می پردازیم و بررسی انجام می دهیم. (۵نمره)
۱۴. در تابع `is_valid_assignment` چک می شود که مقدار تعیین شده برای خانه مد نظر درست هست یا خیر. باید چک شود که در سطر یا ستون یا باکس مربوطه تکرار شده است یا خیر. خروجی به فرمت درست یا نادرست است. (۱۵نمره)
۱۵. در تابع `find_unassigned_location` هم خانه هایی که مقدار برای آن ها تعیین نشده است را مشخص می کنیم. در صورتی که مقدار اولیه تعیین شده بود مقدار ۱-و ۱- برگردانید. (۵نمره)
۱۶. در تابع `solve_csp` با استفاده از روش `csp` و توابع نوشته شده جدول مدنظر حل می شود. (۱۰ نمره)
۱۷. همچنین شما باید در تابع مربوطه محاسبه نمایید که در چند مرحله سودوکو حل شده است. (۵نمره)
۱۸. بهینه شدن کد استفاده شده نیز از بخش های امتیازی پروژه می باشد. (۱۵نمره)
- در توابع `cell` مربوط به `show_Result` با استفاده از توابع نوشته شده در `cell` های قبل توابعی برای ایجاد و حل با دو روش موجود می پردازیم و در نهایت توابع لازمه را فرامیخوانیم.

قوانین:

۱. نمره شما بر اساس گزارش راه طی شده برای حل مسئله و پاسخ صحیح خواهد بود لذا از هرگونه اجتناب در گزارش پرهیز و به موارد خواسته شده به صورت کامل پاسخ دهید.
۲. برای تمرین از شما ارائه شفاهی گرفته خواهد شد بنابراین تسلط لازم را بر کدی که پیاده می کنید داشته باشید.
۳. در صورت مشاهده شباهت غیرعادی بین پیاده سازی ها نمره طرفین طبق قوانین درس محاسبه خواهد شد.
۴. برای تحویل تمرین یک فایل zip شامل گزارش حل سوالات و کدهای نوشته شده، با نام [HW2_SID_NAME] در سامانه LMS بارگذاری کنید.