



تمرین عملی سری اول درس هوش مصنوعی

نام مدرس: دکتر محمدی
دستیاران آموزشی مرتبط: عرفان کلندی، مبین برفی

مهلت تحویل: 5 آبان

بازی KenKen

«کن کن» (KenKen) (یا با این نام دیگر: calcudoku) نوع جدیدی از بازی‌های جدولی است که از جهاتی شبیه به سودوکو می‌باشد، ولی روال متفاوت و ریاضی وارتری دارد. این بازی نخستین بار توسط یک معلم و ریاضیدان ژاپنی به نام «نتسویا میاموتا» معرفی شد. «کن» در فرهنگ ژاپنی به معنای علم و دانش است و کن کن با عنوان مربع دانایی معرفی شده است.

صفحه بازی «کن کن» مثل سودوکو از یک جدول مربع شکل است. معمولاً هر چه تعداد خانه‌های جدول بیشتر باشد حل پازل مشکل‌تر می‌شود. جدول‌های ۴ در ۴ و ۵ در ۵ جدول‌هایی معمول برای بازی هستند. جدول بازی از چند قفس تشکیل شده است. هر قفس شامل چند خانه (cell) از جدول می‌باشد. به طور خلاصه، شما باید خانه‌های هر ناحیه را با اعدادی پر کنید که با به کار بردن عملگر مشخص شده برای آن جدول به جوابی برابر با عددی که در مشخصه داده شده برسید.

قوانین KenKen

1. اعداد منحصر به فرد: هر سطر و هر ستون باید شامل اعداد منحصر به فرد از ۱ تا اندازه‌ی جدول باشد. به عنوان مثال، در یک جدول 4×4 ، اعداد ۱، ۲، ۳ و ۴ باید دقیقاً یک بار در هر سطر و ستون ظاهر شوند.

2. قفس‌ها (Cages): جدول به چندین "قفس" غیر همپوشان (هیچ خانه مشترکی با یک دیگر ندارند) تقسیم شده است. هر قفس با یک عدد هدف و یک عملیات ریاضی (جمع +، تفریق -، ضرب * یا تقسیم /) مشخص می‌شود. اعداد داخل هر قفس باید به گونه‌ای با هم ترکیب شوند که به عدد هدف برسند.

○ به عنوان مثال، اگر یک قفس شامل سه خانه باشد با هدف ۶ و عملیات +، خانه‌ها باید با اعداد پر شوند که مجموع آنها ۶ شود. این می‌تواند شامل (۱، ۲، ۳) یا (۲، ۴) باشد.

3. محدودیت‌های عملیاتی: عملیاتی که برای هر قفس تعریف شده است، همچنین محدودیت‌هایی را در مورد چگونگی ترکیب اعداد اعمال می‌کند:

○ جمع: مجموع اعداد باید برابر با عدد هدف باشد.

تفریق: تفاضل مطلق بین اعداد باید برابر با عدد هدف باشد. (معمولاً برای های دارای ۲ خانه استفاده می‌شود)

○ ضرب: حاصل ضرب اعداد باید برابر با عدد هدف باشد.



تمرین عملی سری اول درس هوش مصنوعی

نام مدرس: دکتر محمدی
دستیاران آموزشی مرتبط: عرفان کلندی، مبین برفی

مهلت تحویل: 5 آبان

○ تقسیم: نتیجه تقسیم عدد بزرگتر بر عدد کوچکتر باید برابر با عدد هدف باشد. (معمولا برای های دارای 2 خانه استفاده میشود)

4. هدف: هدف از معما این است که جدول را به گونه‌ای پر کنیم که تمام شرایط برآورده شود و بدین ترتیب یک راه حل معتبر حاصل شود.

برای آشنایی بیشتر با این بازی می‌توانید به این [لینک](#) مراجعه کنید.

2÷	1	2	3	4
1-	2	1	4	2-
	3	4	2	4×
1-	4	3	1	2

حال در این تمرین وظیفه شما تولید یک برنامه برای تولید و حل جدول ریاضی KenKen می‌باشد. این پیاده‌سازی به دو روش مختلف: backtracking و مسائل ارضای محدودیت (CSP) حل باید انجام شود.

مراحل پیاده‌سازی:

ایجاد جدول اولیه:

تابع به نام generate_KenKen پیاده‌سازی کنید که یک جدول KenKen تصادفی با اندازه‌ی مشخص شده تولید کند. این تابع باید:

- یک جدول به اندازه‌ی size x size ایجاد کند که با صفرها مقداردهی اولیه شده است.
- تابع fill_grid_with_numbers و توابع زیرین آن را به شکلی پیاده‌سازی کنید که جدول را با اعداد از ۱ تا size پر کند در حالی که اطمینان حاصل کند که هر سطر و ستون شامل اعداد منحصر به فرد باشد.



تمرین عملی سری اول درس هوش مصنوعی

نام مدرس: دکتر محمدی
دستیاران آموزشی مرتبط: عرفان کلندی، مبین برفی

مهلت تحویل: 5 آبان

برای این کار تابع کمکی `is_safe_to_place` را طوری پیاده سازی کنید که شرط بالا را برای هر عدد بر اساس توضیحات داک و کد چک کند.

همچنین `fill_grid_backtracking` باید به شکل `backtracking` و با استفاده از تابع بالا عملیات پر کردن جدول را انجام دهد

- در مرحله بعد باید قفس های تصادفی برای جدول ایجاد شده تعریف شود که برای راحت تر شدن پیاده سازی شما این بخش (`generate_random_cages`) پیاده سازی شده و شما باید تنها تابع `calculate_target` را پیاده سازی کنید که نیازمند درک کامل تابع `generate_random_cages` می باشد را پیاده سازی کنید.

* درک نحوه عملکرد این تابع بخشی از نمره تمرین شما را تشکیل می دهد.

- و در نهایت قفسه ها و جدول خالی `size x size` که با صفر مقدار دهی شده است را برگردانید.

پیاده سازی بخش KenKen Backtracking Solver

در این بخش باید توابع مربوط به آن را کامل کرده تا با روش `backtracking` بتوانید جدول داده شده را حل کنید.

- در صورت نبود خانه خالی در `find_unassigned_location -1` و `-1` را باز گردانید.

پیاده سازی بخش KenKen Domain Constraint Solver

در این بخش، باید جدول KenKen را با استفاده از `Constraint Satisfaction Problem (CSP)` حل کنید. در این روش، برای هر خانه، دامنه ای از مقادیر ممکن تعریف می شود که براساس محدودیت های موجود (مانند یکتایی مقادیر در سطرها و ستون ها و همچنین قوانین مربوط به قفس ها) بررسی می شود.

با استفاده از توابعی که برای این بخش در نظر گرفته شده است، می توانید مقادیر درست را به هر خانه اختصاص دهید و در صورت وجود تناقض، فرآیند به صورت بازگشتی به مراحل قبل باز می گردد.



تمرین عملی سری اول درس هوش مصنوعی

نام مدرس: دکتر محمدی
دستیاران آموزشی مرتبط: عرفان کلندی، مبین برفی

مهلت تحویل: 5 آبان

از شما خواسته می‌شود که هر دو الگوریتم **KenKen Backtracking Solver** و **KenKen Domain Constraint Solver** را چندین بار (حداقل 2 بار برای هر سائیز بین سائیزهای 4 تا 7) اجرا کنید و زمان اجرای هر کدام را ثبت نمایید. سپس، میانگین و واریانس زمان‌های اجرا را محاسبه و گزارش کنید.

مراحل انجام کار:

1. هر دو الگوریتم را بر روی مجموعه‌ای از ورودی‌های یکسان اجرا کنید.
2. زمان اجرای هر الگوریتم را برای هر تکرار ثبت کنید.
3. میانگین و واریانس زمان‌های اجرا را محاسبه نمایید.
4. نتایج به‌دست‌آمده را به صورت جدول گزارش و تحلیل کنید.

فایل نوت‌بوک را می‌توانید از [اینجا](#) دانلود کنید.