

گزارش تمرین برنامه نویسی ۲

هلیا شمس زاده

۴۰۰۵۲۱۴۸۶

هوش مصنوعی

—

دکتر محمدی

- **used in row**: در این تابع، با گرفتن عدد و ایندکس ردیف، چک می‌کنیم که عدد داده شده در ردیف داده شده وجود دارد یا نه.
- **used in column**: در این تابع با گرفتن عدد و ایندکس ستون، چک می‌کنیم که عدد داده شده در ستون مورد نظر وجود دارد یا نه. چون سودوکو به شکل لیستی از لیست‌ها است، به ازای تمام لیست‌ها باید ایندکس col چک شود. اگر برابر با عدد داده شده بود مقدار true برگردانده می‌شود.
- **used in box**: در خانه 3×3 جدول که مربوط به ردیف و ستون ورودی است، چک می‌کنیم که عدد داده شده استفاده شده یا نه. اگر استفاده شده بود فلگ found، مقدار true گرفته و ریترن می‌شود.
- **is safe**: این تابع چک می‌کند که عدد داده شده در سطر، ستون و خانه 3×3 مربوطه نباشد. اگر بود false ریترن می‌شود.
- **fill box**: یک لیست از اعداد ۱ تا ۹ درست کرده و اعضای آن را رندوم جابجا می‌کنیم. به ازای هر سلول مربوط به آن خانه 3×3 ، اگر عدد مورد بررسی قبلاً به سلول دیگری داده نشده بود، آن عدد را به آن سلول assign می‌کنیم.
- **fill diagonal**: تابع fill box را برای خانه‌های 3×3 قطر اصلی صدا می‌زند.
- **Find unassigned location**: روی سطر و ستون‌های ماتریس پیمایش می‌کند و ایندکس سطر و ستون سلولی که هیچ مقداری به آن داده نشده (مقدارش صفر است) را برمی‌گرداند. اگر هیچ سلول خالی‌ای وجود نداشت، (۱-، ۱-) را برمی‌گرداند.
- **remove elements**: تعداد عضوهایی که می‌خواهیم از جدول حذف شوند را به عنوان ورودی می‌گیرد و به فلگ removing مقدار اولیه true می‌دهد. در یک حلقه for به اندازه تعداد اعضای که می‌خواهیم remove شود، در یک حلقه while که تا زمانی که فلگ removing برقرار باشد، یک سلول رندوم را چک می‌کند. اگر خالی نبود، آن را خالی کرده به removing را False می‌کند تا حلقه while ادامه پیدا نکند.
- **Generate sudoku**: ابتدا یک ماتریس 9×9 خالی تعریف می‌کند، سپس با تابع fill diagonal خانه‌های قطری آن را پر می‌کند، سپس با کمک تابع solve آن را حل می‌کند، سپس تعداد اعضای که باید remove شوند را از ورودی می‌خواند و تابع Remove elements را صدا زده و جدول ناقص را به عنوان خروجی تحویل می‌دهد.

- **Solve sudoku**: این تابع بازگشتی است. ابتدا تابع `find unassigned location` را صدا می‌زند، اگر مقدار آن $(-1, -1)$ بود، یعنی جدول خانه خالی ندارد و بازی تمام شده، پس از تابع خارج می‌شود. اگر نبود، به ازای اعداد ۱ تا ۹ با تابع `is safe` چک می‌کند که اگر برای آن سلول قابل استفاده است، مقدار آن خانه را برابر آن عدد می‌گذارد و تابع `solve sudoku` را دوباره صدا می‌زند. اگر مقدار `call` نهایی برابر `false` شود، به `call` قبلی برگشته و مقدار سلول را به مقدار قبلی برمی‌گرداند. (`backtrack` می‌کند) و به چک کردن بقیه مقادیر می‌پردازند. اگر جواب را پیدا کرد به طور کامل از تابع خارج می‌شود.
- **display grid**: این تابع اعضای ماتریس نهایی را مثل جدول سودوکو چاپ می‌کند.

- **find unassigned location**: این تابع همانند تابع بخش `generate sudoku` سطر و ستون خانه خالی پیدا شده را برمی‌گرداند. اگر خانه خالی وجود نداشت، $(-1, -1)$ برمی‌گرداند.
- **Create domains**: یک دیکشنری (`assignment`) برمی‌گرداند که کلیدهای آن، سطر و ستون سلول‌ها هستند و مقادیر آن برای هر کلید لیستی از دامنه‌هایی است که می‌توان به آن خانه داد. اگر آن سلول خالی نبود، مقدار آن خانه را برای `Value` اش قرار می‌دهد.
- **Is valid assignment**: اگر `val` در سطر و ستون و خانه 3×3 مربوط به آن سلول نبود، `true` برمی‌گرداند.
- **Forward checking**: این تابع مقدار `val`، `row` و `col` را می‌گیرد و مقدار سلول مربوطه را برابر `val` گذاشته و مقدار `val` را از سلول‌هایی که با سلول ورودی `constraint` دارند (یعنی سلول‌های آن سطر، ستون و خانه 3×3 مربوطه) از `domain` شان حذف می‌کند. همچنین، حین حذف کردن دامنه‌ها، در صورت خالی شدن دامنه یک سلول، مقدار `false` برمی‌گرداند. (درروش `fc`، در صورت خالی شدن دامنه یک سلول، `backtrack` انجام می‌شود.) همچنین سلول‌هایی که مقداری از دامنه‌شان حذف شده را درون یک لیست ذخیره می‌کند و برمی‌گرداند.

- **undo forward checking**: این تابع اثرات ناشی از تابع forward checking را خنثی می‌کند تا در صورت backtrack، مقدار حذف شده به دامنه‌ها برگردد.
- **Solve csp**: این تابع یک سلول خالی را با تابع find unassigned location پیدا می‌کند، در صورتی که خانه خالی نبود، خارج می‌شود. سپس به ازای مقادیر دامنه آن سلول، for می‌زند و مقدار آن سلول را برابر با یک مقدار از دامنه‌اش که Safe است قرار داده و تابع solve csp را دوباره صدا می‌زند. اگر تابع solve csp صدا زده شده نهایتاً به جواب نرسید، اثرات این تابع را با undo forward checking خنثی کرده (backtrack) و به چک کردن بقیه مقادیر می‌پردازد.