



دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)

دانشکده ریاضی و علوم کامپیوتر

استاد درس: دکتر مهدی قطعی

استاد کارگاه: بهنام یوسفی مهر

پاییز ۱۴۰۲

راهنمای تمرین ۴

درس هوش مصنوعی و کارگاه



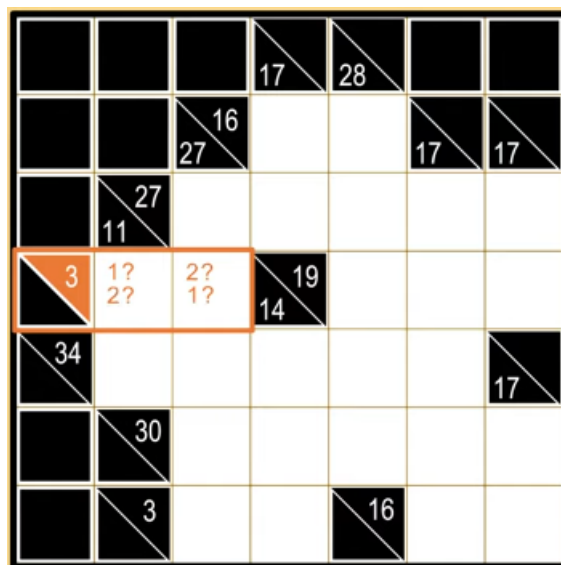
آشنایی با کاکورو

کاکورو^۱ یکی از پازل‌هایی است که شباهت زیادی به جدول کلمات متقاطع^۲ دارد. تفاوت این دو پازل در این است که در کاکورو به جای حروف با ارقام سر و کار داریم. در جدول کلمات متقاطع راهنماهایی وجود دارند که به کمک آنها حروف را در ستون‌ها و یا سطرها قرار می‌دهیم. اما در کاکورو جمع اعداد مربوط به بخشی از یک سطر یا ستون به ما داده شده و باید ارقام ۱ تا ۹ را به گونه‌ای در هر بخش قرار دهیم که جمع آنها با جمع مطلوب آن بخش برابر باشد. جمع مطلوب هر بخش را راهنما^۳ می‌گویند. شرط دیگری که باید رعایت شود این است که ارقامی که با هم جمع می‌شوند تا به یک راهنما برسند نباید تکراری باشند. برای فهم بهتر این پازل و قوانین آن به مثال زیر توجه کنید.

			17	28		
		16			17	17
	27					
3			19			
34			14			17
	30					
	3			16		

یکی از استراتژی‌های مرسوم برای حل دستی این پازل‌ها شروع از راهنماها با مقادیر خیلی کم یا خیلی زیاد است که حالت‌های ممکن کمتری دارند. به عنوان مثال در اینجا از راهنمای برابر با ۳ شروع می‌کنیم و حدس می‌زنیم که در هر خانه‌ی مربوط به آن راهنما چه ارقامی می‌توانند قرار بگیرند. چون ۳ عدد کوچکی است، تنها حالت برای جمع دو عدد متمایز برای اینکه ۳ شود برابر ۱ و ۲ است که البته در ابتدا نمی‌دانیم ۱ یا ۲ هر کدام در کدام خانه قرار می‌گیرند. پس حالت‌های ممکن را در خانه‌های مربوطه نشان می‌کنیم.

^۱ Kakuro^۲ Crossword puzzle^۳ clue



در ادامه راهنمای ۱۱ را انتخاب کرده و می‌دانیم حالت‌های زیر برای دو خانه‌ی مربوط به این راهنما وجود دارد:

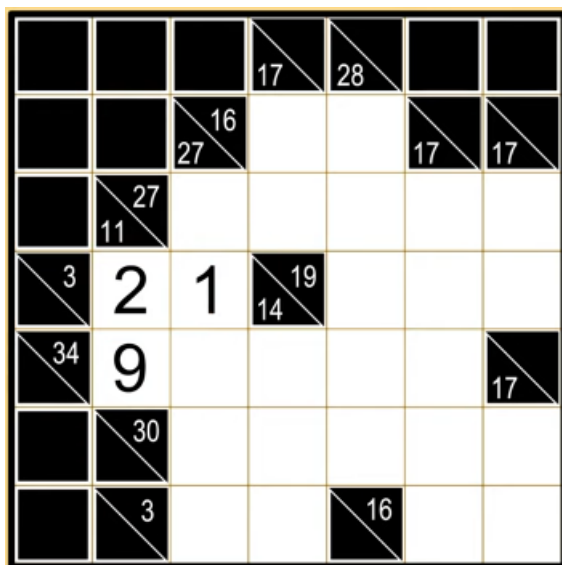
۱. $۹ + ۲$

۲. $۸ + ۳$

۳. $۷ + ۴$

۴. $۶ + ۵$

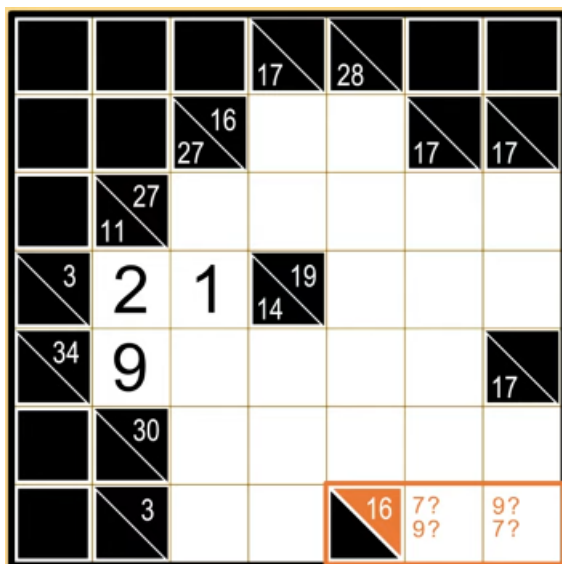
اما همان طور که از مرحله‌ی قبل نشان کرده بودیم یکی از خانه‌ها باید ۲ یا ۱ باشد، پس تنها حالت قابل قبول حالت اول است و لذا تا اینجا کار پازل به شکل زیر به دست آمده است.



به همین صورت می‌توانیم حل را ادامه دهیم. اما باید توجه داشت که ارقام تکراری نباید در خانه‌های یک راهنما قرار بگیرند. برای مثال اگر تصمیم بگیریم به سراغ راهنمای ۱۶ برویم، آنگاه از بین دو حالت زیر فقط حالت اول قابل قبول است:

۱. $9 + 7$

۲. $8 + 8$





پیاده‌سازی عامل هوشمند حل‌کننده‌ی کاکورو

بخش اول: روش پس‌گرد

قبل از اینکه بخواهیم عامل هوشمند بازی را طراحی کنیم بهتر است یک عامل ساده طراحی کرده و به مرور این عامل را بهبود بدهیم. یکی از ساده‌ترین عامل‌ها عاملی است که با الگوریتم پس‌گرد^۴ پازل را حل می‌کند. برای آشنایی با این الگوریتم می‌توانید **فصل دوم** کتاب Algorithms اثر Jeff Erickson را مطالعه کنید. در این بخش انتظار می‌رود با استفاده از این الگوریتم عامل اولیه‌ی کاکورو را پیاده‌سازی کنید. توجه کنید که در بخش نهایی از این عامل استفاده خواهید کرد پس این عامل را ذخیره نگه دارید.

بخش دوم: بهبود روش پس‌گرد

در گام بعدی می‌خواهیم عامل را هوشمند کنیم. عاملی که در بخش اول طراحی می‌کنیم درخت جست‌وجویی را در بین فضای حالات پازل طی می‌کند تا به پاسخ درست برسد. پیمایش چنین درختی مشابه الگوریتم DFS است. حال از خود سوال می‌کنیم که آیا عامل فقط یک درخت منحصر بفرد را می‌تواند جست‌وجو کند؟ آیا نمی‌توان شکل درخت جست‌وجو را بهبود بخشید؟ آیا نمی‌توان پیمایش عامل روی درخت جست‌وجو را بهبود داد؟ در مثالی که مطرح کردیم دیدیم شروع منطقی زمانی است که از راهنماهای با مقادیر کم یا مقادیر زیاد شروع کنیم. آیا می‌توانید ربط این کار با هیوریستیک MCV^۵ را ببینید؟ همچنین پس از راهنمای ۳ سراغ راهنمای ۱۱ رفتیم. دلیل این عمل ما وجود خانه‌ی مشترک میان راهنمای ۳ و راهنمای ۱۱ بود. حال سوال اینجاست که آیا می‌توان منطقی‌تر ترتیب پیمایش راهنماها را در الگوریتم پس‌گرد انجام داد؟

انتظار می‌رود در این قسمت روش پس‌گرد را با الگوریتم‌های CSP^۶ مطرح شده در کلاس بهبود داده و در هر بهبود عامل خود را ذخیره کرده و در بخش بعدی آن‌ها را با هم مقایسه کنید. در این بخش می‌توانید از هیوریستیک‌های MCV و یا LCV^۷ برای بهبود و افزایش سرعت بیشتر استفاده کنید و عامل‌های ترکیبی با این الگوریتم‌ها و هیوریستیک‌ها طراحی نمایید.

بخش سوم: مقایسه‌ی عامل‌ها

در این قسمت باید عامل‌های خود در بخش‌های اول و دوم را با هم مقایسه کرده و گزارشی از نحوه‌ی عملکرد هر کدام ارائه کنید. برای مقایسه می‌توانید عامل‌های خود را روی یکی از هر چهار دسته‌ی پازل‌های **اینجا** امتحان کنید.

^۴backtracking

^۵Most Constrained Variable

^۶Constraint Satisfaction Problem

^۷Least Constraining Value



معیارهای سنجش تمرین

- در ابتدا بازی کاکورو را پیاده‌سازی کنید و توضیحی از نحوه‌ی پیاده‌سازی و منابع استفاده شده بدهید. استفاده از کدها و پیاده‌سازی‌های آماده در این بخش با ذکر منبع بلامانع است.
- در بخش‌های مختلف مطابق همین راهنما عامل هوشمند کاکورو را پیاده‌سازی کرده و با مقایسه و تحلیل عامل‌ها، نتیجه‌گیری خود را از تمرین بیان کنید. در قسمت مقایسه تاثیر هر بهبود در عامل را توجیه کنید. بهترین عامل شما باید در زمان معقولی پازل دلخواهی از کاکورو را به درستی حل کند. همچنین توجه کنید که پیاده‌سازی عامل‌های حل‌کننده‌ی پازل در این قسمت باید بدون استفاده از کدهای آماده و کاملاً توسط خود شما انجام شود.

نحوه‌ی پیوست کد به گزارش

اگر کدتان را در Google Colab پیاده‌سازی کرده‌اید لینک آن را در انتهای گزارش اضافه کنید. همچنین دسترسی به کد از طریق لینک را نیز در فایل Colab خود فعال کنید. می‌توانید کد را در Github بارگذاری کرده و لینک Github را به انتهای گزارش اضافه نمایید.

مهلت تمرین

برای ارسال این تمرین تا پایان روز جمعه ۳ آذر فرصت دارید. لطفاً فایل نهایی را با قالب مشخص شده و فقط در سامانه‌ی کورسز بارگذاری نمایید.