جواب کوییز2 هوش مصنوعی

1.چهار مرحله کلی برای حل یک مساله در هوش مصنوعی را با مثال شهر رومانی شرح دهید:

حل یک مسئله در هوش مصنوعی معمولاً به چندین مرحله تقسیم می‌شود. در زیر، چهار مرحله کلی برای حل یک مساله در هوش مصنوعی را با مثال شهر رومانی شرح می‌دهم:

تعریف مسئله: در این مرحله، مسئله به طور دقیق تعریف می‌شود. باید مشخص شود که مسئله دقیقاً چه هدفی دارد و چه محدودیت‌هایی دارد. به عنوان مثال، می‌توانیم مسئله مسیریابی در شهرهای رومانی را در نظر بگیریم که هدف آن پیدا کردن کوتاه‌ترین مسیر بین دو شهر است و محدودیت‌ها شامل مسافت‌های بین شهرها و محدودیت‌های مربوط به طول و زمان سفر است. طراحی مدل: در این مرحله، باید یک مدل ریاضی برای حل مسئله طراحی کنیم. این مدل معمولاً با استفاده از مفاهیم ریاضی و الگوریتم‌های مختلف ایجاد می‌شود. در مثال مسیریابی در شهرهای رومانی، می‌توانیم از یک گراف استفاده کنیم که شهرها را به عنوان گره‌ها و مسافت‌های بین شهرها را به عنوان یال‌ها نشان می‌دهد. سپس با استفاده از الگوریتم‌های جستجو، می‌توانیم کوتاه‌ترین مسیر را پیدا کنیم. پیاده‌سازی و اجرا: در این مرحله، مدل طراحی شده را پیاده‌سازی می‌کنیم و بر روی داده‌های واقعی اجرا می‌کنیم. می‌توانیم از زبان برنامه‌نویسی مانند Python یا Java برای پیاده‌سازی استفاده کنیم. در مثال مسیریابی در شهرهای رومانی، باید گراف و مسافت‌های بین شهرها را به صورت داده‌های واقعی وارد کنیم و الگوریتم مسیریابی را بر روی آنها اجرا کنیم. ارزیابی و بهبود: در این مرحله، عملکرد مدل و الگوریتم را ارزیابی می‌کنیم و در صورت نیاز، آنها را بهبود می‌بخشیم. این مرحله ممکن است نیاز به تجزیه و تحلیل داده‌های خروجی و تغییر در طراحی مدل داشته باشد. به عنوان مثال، می‌توانیم در مساله مسیریابی، عملکرد الگوریتم را با تغییر پارامترها یا استفاده از الگوریتم‌های بهتر بهبود دهیم. 2.انواع مساله را نام ببرید و شرح مختصری از هر یک با ذکر یک مثال بیان کنید؟

مساله جستجو و کاوش: در این نوع مساله، هدف پیدا کردن یک راه‌حل مناسب از میان یک فضای جستجو است. مثالی از این نوع مسئله می‌تواند جستجوی مسیر کوتاه‌ترین بین دو شهر در یک نقشه باشد. مساله بهینه‌سازی: در این نوع مساله، هدف پیدا کردن بهترین راه‌حل بر اساس یک معیار است. به عنوان مثال، مساله برنامه‌ریزی تولید بهینه برای توزیع محصولات به فروشگاه‌ها. مساله تصمیم‌گیری: در این نوع مساله، هدف انتخاب بهترین اقدام بر اساس ورودی‌های داده شده است. به عنوان مثال، مساله تصمیم‌گیری برای خرید سهام بر اساس اطلاعات بازار و تحلیل ریسک. مساله پردازش زبان طبیعی: در این نوع مساله، هدف درک و تفسیر زبان طبیعی توسط سامانه‌های مصنوعی است. به عنوان مثال، سیستم‌های تشخیص گفتار و ترجمه ماشینی. مساله یادگیری ماشین: در این نوع مساله، هدف آموزش سامانه‌های مصنوعی برای یادگیری از داده‌ها و بهبود عملکرد آنها است. به عنوان مثال، سیستم توصیه‌گر بر اساس سابقه خرید کاربران. مساله شناخت تصویر: در این نوع مساله، هدف تشخیص و تفسیر تصاویر توسط سامانه‌های مصنوعی است. به عنوان مثال، سیستم تشخیص چهره در عکس‌ها. مساله منطق و استدلال: در این نوع مساله، هدف استدلال بر اساس قواعد و منطق است. به عنوان مثال، سیستم‌های تشخیص تقلب در مالیات بر اساس الگوریتم‌های منطقی. این تنها چند نمونه از انواع مختلف مسائل در هوش مصنوعی است و هر کدام از این مسائل به روش‌ها و الگوریتم‌های مختلفی نیاز دارند. 3.مسئله 8 وزیر را با دو روش فرموله سازی کنید(مثالn وزیر را طوری در صفحه شطرنج بگذارید که همدیگر را تهدید نکنند)؟

روش اول: استفاده از مساله فرمولاسیون CSP (محدودیت برنامه‌ریزی) یک روش برای فرموله‌سازی این مسئله استفاده از فرمولاسیون CSP (محدودیت برنامه‌ریزی) است. در این روش، هر خانه در صفحه شطرنج به عنوان یک متغیر در نظر گرفته می‌شود و مقدار هر متغیر می‌تواند عدد 1 تا N (تعداد وزیرها) باشد. سپس با استفاده از محدودیت‌ها، تهدیدهای ممکن بین وزیرها را مدل می‌کنیم. برای مثال، محدودیتی می‌تواند باشد که دو وزیر نباید در یک ستون، یک سطر یا قطری باشند. روش دوم: الگوریتم ژنتیک روش دیگری برای فرموله‌سازی این مسئله استفاده از الگوریتم ژنتیک است. در این روش، هر حالت (موقعیت وزیرها در صفحه شطرنج) را با یک کروموزوم و هر ویژگی (موقعیت هر وزیر) را با یک ژن نمایش می‌دهیم. سپس با استفاده از عملگرهای تکاملی مانند انتخاب، ترکیب، و جهش، بهترین حالت‌ها را پیدا کرده و تا جای ممکن تهدیدها را حذف می‌کنیم. به این ترتیب، ترکیبی از وزیرها را در صفحه شطرنج بدون تهدیدها پیدا می‌کنیم. برای مثال، فرض کنید بخواهیم 8 وزیر را در صفحه شطرنج 8x8 قرار دهیم. این مسئله به این صورت فرموله می‌شود:

هر وزیر می‌تواند در یک سطر از 1 تا 8 قرار بگیرد.

هر وزیر می‌تواند در یک ستون از 1 تا 8 قرار بگیرد. هر وزیر می‌تواند در یک قطر اصلی یا فرعی از 1 تا 8 قرار بگیرد. جستجوی درختی را با ذکر یک مثال شرح دهید؟

در این روش، یک درخت جستجوی ساخته می‌شود که هر گره آن یک حالت ممکن در مسئله را نمایش می‌دهد و یال‌های درخت نشان‌دهنده روابط بین حالت‌ها هستند. هدف جستجوی درختی پیدا کردن مسیری است که به یک حالت هدف منتهی می‌شود. برای مثال، فرض کنید می‌خواهیم یک مسئله جستجوی کوتاه‌ترین مسیر در یک نمودار جهانی حل کنیم. در این مسئله، هر گره از درخت جستجو یک شهر را نمایش می‌دهد و یال‌ها نشان‌دهنده مسیرهای ممکن بین شهرها هستند. همچنین، هر گره شامل اطلاعاتی مانند مسافت تا شهرهای دیگر و هزینه سفر است. با شروع از گره شهر مبدأ، می‌توانیم با استفاده از الگوریتم جستجوی درختی مانند الگوریتم جستجوی سطح اول (BFS) یا الگوریتم جستجوی عمق اول (DFS)، درخت جستجو را پیمایش کنیم. هر بار که به یک گره جدید می‌رسیم، بررسی می‌کنیم که آیا این گره حالت هدف است یا خیر. اگر گره حالت هدف باشد، مسیری که تا این گره رسیده‌ایم را پیدا کرده‌ایم. در غیر این صورت، ادامه می‌دهیم و به تمام گره‌های فرزند این گره می‌رویم و جستجو را در آنها ادامه می‌دهیم. به عنوان مثال، فرض کنید می‌خواهیم کوتاه‌ترین مسیر بین شهر A و شهر B را در یک نمودار جهانی پیدا کنیم. درخت جستجوی ساخته شده از شهر A شروع می‌شود و با پیمایش گره‌های فرزند، به تمام شهرهای ممکن در نمودار می‌رسیم. هر بار که به گره شهر B می‌رسیم، مسیری که تا اینجا پیموده‌ایم را پیدا کرده‌ایم. این مسیر می‌تواند از تعدادی شهر و مسیرهای مختلف تشکیل شده باشد که با استفاده از جستجوی درختی، به آن دست پیدا می‌کنیم.

5.فضای حالت و Fringe را تعریف کنید؟

فضای حالت (State Space) در مفهوم عمومی، مجموعه‌ای است از تمام حالت‌های ممکن که یک سامانه می‌تواند به آن‌ها وارد شود یا در آن‌ها قرار بگیرد. در حوزه هوش مصنوعی و جستجوی مسئله، فضای حالت به معنای مجموعه‌ای از تمام حالت‌های ممکن است که یک مسئله را می‌توان با آن‌ها مدل کرد. هر حالت در فضای حالت می‌تواند یک توصیف کامل از وضعیت سامانه در یک زمان خاص باشد. به عنوان مثال، در یک مسئله جستجوی مسیر، هر حالت می‌تواند مکان واقعی یک عامل در یک زمان خاص باشد. Fringe یا لبه (Fringe) در جستجوی مسئله، مجموعه‌ای است از گره‌ها که در حالت‌هایی قرار دارند که هنوز بررسی نشده‌اند و در مراحل بعدی ممکن است بررسی شوند. به عبارت دیگر، Fringe شامل گره‌هایی است که در حالت‌های مجاور به حالت‌های فعلی قرار دارند. در جستجوی مسئله، از Fringe برای نگهداری گره‌هایی که باید در مراحل بعدی بررسی شوند و به آن‌ها دسترسی داشته باشیم، استفاده می‌شود. در هر مرحله از جستجو، یک گره از Fringe برای بررسی انتخاب می‌شود و به عنوان گره فعلی در نظر گرفته می‌شود. به عنوان مثال، در جستجوی کوتاه‌ترین مسیر بین دو شهر، فضای حالت شامل تمام شهرهای ممکن است که می‌توان به آن‌ها رسید. Fringe در این مسئله مجموعه‌ای از گره‌هایی است که در حالت‌های مجاور به حالت فعلی قرار دارند و هنوز بررسی نشده‌اند. در هر مرحله، یک گره از Fringe انتخاب می‌شود و بررسی می‌شود که آیا به حالت هدف رسیده است یا خیر. اگر به حالت هدف رسیده باشد، مسیری که تا این گره طی شده است، پیدا می‌شود. در غیر این صورت، گره‌های فرزند این گره به Fringe اضافه می‌شوند تا در مراحل بعدی بررسی شوند.

6.جستجوی ناآگاهانه را تعریف کنید و انواع آن را نام ببرید؟

جستجوی ناآگاهانه (Uninformed Search) یک روش جستجو در هوش مصنوعی است که در آن الگوریتم‌ها بر اساس اطلاعات کمتری درباره فضای حالت و مسئله عمل می‌کنند. در جستجوی ناآگاهانه، الگوریتم‌ها بر اساس قوانین عمومی جستجو عمل می‌کنند و به صورت سیستماتیک و بدون اطلاعات خاص درباره مسئله، حالات را بررسی می‌کنند. انواع جستجوی ناآگاهانه عبارتند از: جستجوی سطح اول (Breadth-First Search): در این روش، ابتدا گره‌های مجاور به گره فعلی بررسی می‌شوند و سپس به گره‌های مجاور به آن‌ها می‌رسیم. این جستجو به صورت وسیع و پهن از درخت جستجو پیش می‌رود و تمام گره‌های هر سطح را در ابتدا بررسی می‌کند. جستجوی عمق اول (Depth-First Search): در این روش، ابتدا یک گره را بررسی می‌کنیم و سپس به گره فرزند آن می‌رسیم. این روش به صورت عمقی در درخت جستجو پیش می‌رود و تا زمانی که دیگر گره‌ای برای بررسی نباشد، به صورت پیوسته به سمت پایین حرکت می‌کند. جستجوی یکنواخت (Uniform-Cost Search): در این روش، هر گره به عنوان نقطه شروع در نظر گرفته می‌شود و هزینه حرکت به هر گره در نظر گرفته می‌شود. الگوریتم در هر مرحله گره با کمترین هزینه را بررسی می‌کند و به سمت آن حرکت می‌کند.

جستجوی عمق محدود (Limited-Depth Search): در این روش، محدودیتی برای عمق جستجو تعیین می‌شود و تا آن حد عمق، جستجو انجام می‌شود. این روش می‌تواند به صورت محدودیت زمانی یا محدودیت عمق باشد. جستجوی عمق تدریجی (Iterative Deepening Search): این روش ترکیبی از جستجوی عمق اول و جستجوی عمق محدود است. در ابتدا با عمق یک شروع می‌کند و در هر مرحله عمق را افزایش می‌دهد تا به حل مسئله برسد. 7.الگوریتمی که از لحاظ زمانی از مرتبه جستجوی اول سطح است ولی از لحاظ پیچیدگی حافظه از مرتبه جستجوی اول عمق می باشد کدام است ،شرح دهید؟

الگوریتمی که از لحاظ زمانی از مرتبه جستجوی اول سطح (Breadth-First Search) است ولی از لحاظ پیچیدگی حافظه از مرتبه جستجوی اول عمق (Depth-First Search) می‌باشد، الگوریتم جستجوی بهینه‌شده (Optimized Search) نامیده می‌شود. در الگوریتم جستجوی بهینه‌شده، از هر دو روش جستجوی اول سطح و جستجوی عمق اول بهره‌برداری می‌شود. الگوریتم به این صورت عمل می‌کند که ابتدا با استفاده از جستجوی اول سطح، گره‌های مجاور به گره فعلی را بررسی می‌کند و در صورتی که به هدف نرسیده باشد، به گره‌های مجاور به آن‌ها می‌رسد. این روند تا زمانی ادامه می‌یابد که به یک عمق مشخصی برسد. در این مرحله، الگوریتم به جستجوی عمق اول تغییر می‌دهد و از روش جستجوی عمق اول برای بررسی گره‌های مجاور به گره‌های فعلی استفاده می‌کند. این کار ادامه می‌یابد تا زمانی که به هدف برسد یا تمام گره‌ها را بررسی کند. با استفاده از این الگوریتم، زمان اجرای جستجوی اول سطح را داریم که به صورت وسیع و پهن از درخت جستجو پیش می‌رود و تمام گره‌های هر سطح را در ابتدا بررسی می‌کند. اما با تغییر به جستجوی عمق اول، پیچیدگی حافظه نیز از مرتبه جستجوی اول عمق می‌شود، زیرا در الگوریتم جستجوی عمق اول، تنها یک مسیر را در حافظه نگه می‌دارد و در صورت نرسیدن به هدف، به گره‌های فرزند آن می‌رود.

8.کارایی انواع جستجوهای نا آگاهانه را بر حسب چهار پارامتر کامل بودن ، بهینگی ، پیچیدگی زمانی و فضایی بیان کنید؟

انواع جستجوهای نا آگاهانه به ترتیب کامل بودن ، بهینگی ، پیچیدگی زمانی و پیچیدگی فضایی مورد بررسی قرار می‌گیرند:

کامل بودن (Completeness): یک الگوریتم جستجوی نا آگاهانه کامل است اگر در صورت وجود پاسخ، همواره به پاسخ برسد. به عبارت دیگر، الگوریتم باید تمام مسیرهای ممکن را بررسی کند تا به پاسخ برسد. الگوریتم جستجوی سطح اول (Breadth-First Search) و جستجوی عمق اول (Depth-First Search) از نوع جستجوهای نا آگاهانه هستند و برای مسائل محدود شده کامل هستند. بهینگی (Optimality): یک الگوریتم جستجوی نا آگاهانه بهینه است اگر در صورت وجود پاسخ، به پاسخ بهینه برسد. به عبارت دیگر، الگوریتم باید بهترین پاسخ ممکن را ارائه کند. الگوریتم جستجوی سطح اول و جستجوی عمق اول بهینه نیستند، زیرا ممکن است به پاسخ غیر بهینه برسند. پیچیدگی زمانی (Time Complexity): پیچیدگی زمانی الگوریتم جستجوی نا آگاهانه به تعداد عملیاتی که در طول اجرا انجام می‌دهد بستگی دارد. الگوریتم جستجوی سطح اول و جستجوی عمق اول هر دو از مرتبه O(b^d) هستند، که در آن b تعداد حالت‌های ممکن در هر گره و d عمق مسئله را نشان می‌دهد. الگوریتم جستجوی سطح اول معمولاً زمان بیشتری نسبت به جستجوی عمق اول می‌برد زیرا تمام گره‌های هر سطح را در ابتدا بررسی می‌کند. پیچیدگی فضایی (Space Complexity): پیچیدگی فضایی الگوریتم جستجوی نا آگاهانه به حافظه‌ای که در طول اجرا استفاده می‌شود بستگی دارد. الگوریتم جستجوی سطح اول به مقدار O(b^d) حافظه نیاز دارد زیرا تمام گره‌های هر سطح را در حافظه نگه می‌دارد. الگوریتم جستجوی عمق اول به مقدار O(bd) حافظه نیاز دارد زیرا تنها یک مسیر را در حافظه نگه می‌دارد.

ت مواجه می شود؟ راهکارهای پیشنهادی خود را نام برده و مختصری در خصوص هر کدام توضیح دهید؟ عامل مبتنی بر جدول به شکست می‌خورد زیرا در این حالت، عامل فقط بر اساس تاریخچه‌ی مشاهدات قبلی خود عمل می‌کند و هیچ گونه فرایند تصمیم‌گیری یا یادگیری دینامیک ندارد. این در حالی است که در بسیاری از مسائل واقعی، محیط پویا و تغییرپذیر است و عامل باید قادر به انطباق با تغییرات محیط باشد. برای رفع این مشکل و ارتقاء عامل مبتنی بر جدول، می‌توان از راهکارهای زیر استفاده کرد: استفاده از الگوریتم‌های یادگیری تقویتی: الگوریتم‌های یادگیری تقویتی، مانند Q-learning و SARSA، عامل را قادر به یادگیری راه‌حل بهینه برای مسئله می‌کنند. این الگوریتم‌ها بر اساس مفهوم پاداش و تنبیه عمل می‌کنند و با بهبود تصمیم‌گیری عامل، عملکرد آن را بهبود می‌بخشند. استفاده از شبکه‌های عصبی: شبکه‌های عصبی قادر به تعمیم داده‌ها و یادگیری الگوهای پیچیده هستند. با استفاده از این روش، عامل قادر به تصمیم‌گیری در مورد وضعیت‌های مختلف می‌شود و می‌تواند با تغییرات محیط سازگار شود. استفاده از الگوریتم‌های تصمیم‌گیری مارکوف: الگوریتم‌های تصمیم‌گیری مارکوف (MDP) با استفاده از تئوری احتمال و تصمیم‌گیری بهینه، عامل را در تصمیم‌گیری‌های پیچیده راهنمایی می‌کنند. این الگوریتم‌ها مبتنی بر مدلی از محیط هستند و با استفاده از توابع ارزش و عملکرد، به عامل کمک می‌کنند تا راه‌حل بهینه را پیدا کند. با استفاده از این راهکارها، عامل مبتنی بر جدول قادر خواهد بود با محیط پویا و تغییرپذیر تعامل کند و عملکرد بهتری داشته باشد. \\\\\\\\\\\\\\\\\\\\\\\\\ 6دنیای جاروبرقی راباتوجه به فرموله سازی مسئله تشریح کنید؟ حالتها) states ):حالت به وسیله مکان عامل ومکانهای کثیف تعیین میشود.عامل دریکی ازدومکان است که هرکدام ممکن است کثیف باشندیانباشند.پس8=2^ 2\*2 حالت وجوددارد.یعنی)n2.^n)حالت. حالت شروع:هرحالتی میتواندبه عنوان حالت شروع باشد. فعالیتها)action):دراین محیط جاروبرقی فقط سه فعالیت میتواندانجام دهد1:(حرکت به سمت چپ 2(حرکت به سمت راست 3(عمل مکش آزمون هدف:بررسی میکنذآیاتمام مکانهاتمیزاست یاخیر. هزینه ی مسیر:تعدادمراحل موجوددرمسیر،هزینه ی مسیراست. \\\\\\\\\\\\\\\\\\\\\\\\\ 7 جستجوی عمقی را با رسم مرحله به مرحله شرح دهید و در نهایت کارایی الگوریتم را با چهار معیاراندازی گیری بیان کنید؟ جستجوی عمقی (Depth-First Search) یک الگوریتم جستجو در گراف است که با استفاده از استک به صورت عمق به جستجوی گره هدف می‌پردازد. در این الگوریتم، عملیات جستجو از ریشه شروع می‌شود و به صورت پیوسته به سمت پایین درخت جستجو حرکت می‌کند تا به گره هدف برسد یا دیگر قابلیت حرکت نداشته باشد. در هر مرحله، یک گره از استک خارج شده و اعمال ممکن در آن گره بررسی می‌شود. در صورتی که گره هدف یافت شود، جستجو متوقف می‌شود و در غیر این صورت، جستجو به عمق بیشتر ادامه می‌یابد. مراحل جستجوی عمقی به صورت زیر است: قرار دادن گره شروع در استک. تکرار این مراحل تا زمانی که استک خالی شود: گرفتن گره فعلی از استک. بررسی آیا گره فعلی گره هدف است یا خیر. در صورت برابر بودن گره فعلی با گره هدف، جستجو متوقف می‌شود و مسیر به گره هدف برگشت داده می‌شود. در غیر این صورت، تمام فرزندان گره فعلی را به استک اضافه می‌کنیم. در صورتی که استک خالی شود و گره هدف پیدا نشود، جستجو به نتیجه "ناموفق" می‌رسد. کارایی الگوریتم جستجوی عمقی می‌تواند با استفاده از چهار معیار اندازه‌گیری شود: زمان اجرا: زمانی که الگوریتم برای پیدا کردن گره هدف نیاز دارد. حافظه: میزان حافظه مورد نیاز برای ذخیره و مدیریت استک و گره‌های جستجو شده. پیچیدگی زمانی: تعداد گره‌ها و یال‌هایی که باید بررسی شوند تا به گره هدف برسیم. کیفیت جواب: میزان بهینگی جواب‌هایی که الگوریتم جستجوی عمقی به دست می‌دهد. با توجه به معیارهای فوق، الگوریتم جستجوی عمقی معمولاً در مسائلی که گراف جستجو از اندازه بزرگی برخوردار است و تعداد عمق جستجو کم است، کارایی خوبی دارد. اما در مسائلی که عمق جستجو بیشتر است و ممکن است در درخت جستجو به گره‌های تکراری برخورد کند، ممکن است به شکست بخورد و کارایی پایینی داشته باشد. \\\\\\\\\\\\\\\\\\\\\\\\\ 7 جست و جوی عمقی را با رسم مرحله به مرحله شرح دهید و در نهایت کارایی الگوریتم را با چهار معیار اندازه گیری بیان کنید ؟ جستجوی عمقی ،عمیق ترین گره رابسط میدهد،جستجوازعمیق ترین سطح درخت جستجوادامه می یابد،وقتیوقتی گره ها بسط داده شدند ازمرزحذف میشوندوجستجوبه عمیق تری ن گره بعدی برمی گردد.جستجوی عمقی ازصف LIFO استفاده میکند.دراین صف جدیدترین گره تولیدشده،برای بسط دادن انتخاب میشود،این گره بایدعمیق ترین گره بسط نداده شده باشد

جستجوی عمقی: کامل بودن : خیر ، مگر اینکه فضای حالت محدود باشد و حلقه تکرار وجود نداشته باشد . بهینه بودن : خیر ، چون کامل نیست . پیچیدگی زمانی(m^b(O ، اگر m خیلی بزرگتر از d باشد به مراتب بدتر است / در بسیاری از مسائل سریعتر از جست وجوی BF است . پیچیدگی حافظه (+1bm(O : ، در زمان عقبگرد حافظه آزاد می شود . \\\\\\\\\\\\\\\\\\\\\\\\\ 8 ضمن بررسی الگوریتم جستجوی درختی شبه کد زیر را بررسی کنید که استراتژی در کدام از 4 توابع ، پیاده سازی شده است ، توابع را نام برده و عملکرد هر یک را بیان کنید درالگوریتم جستجوی درختی ،حالت شروع درریشه درخت قرارمی گیرد،انشعابها،فعالیتها و گره ها،حالتهای موجودهستند.ابتداریشه رابررسی میکنیم که ایاحالتهدف است یاخیر درصورتی که حالت هدف نبودان رابسط میدهیم تامجموعه ی جدیدی ازحالتهابه وجودآید،بعدازآن حالتهارایکی یکی بررسی کرده تازمانی که به آخرین گره برسیم که هیچ فرزندی ندارد.پس سراغ گره ها میرویم ویکی یکی بررسی میکنیم پس ازآن گره هایی که مارا به هدف نمیرساندحذف میکنیم واین روش ادامه پیدامیکندتابه هدف برسیم. استراتژی های متفاوتی برای رسیدن به حالت هدف وجوددارد.استراتژی مادراینجااین است که یک گره کاندید رابررسی کن اگر هدف نبود آن رابسط بده،آنقدراین کارراتکرار کن تا به هدف برسی. تابع)first remove):اولین خانه را fringeمیکند تابع)test goal):آیابه هدف رسیدیم؟خیر.یک گره باتوجه به استراتژی انتخاب کن تابع)expand):وقتی به هدف نرسیدیم گره هارابسط بده. تابع)insert):گره های فرزندرادرfringبسط بده ونتایج رابه جستجواضافه کن. استراتژی در تابع insert پیاده سازی شده است. \\\\\\\\\\\\\\\\\\\\\\\\\ 9 شبه کد زیر مربوط به کدام جست و جوی ناآگاهانه می باشد ، از مزایای کدام جست و جو های دیگر بهره برده است ، با ترسیم شکل توضیح دهید ؟ این شبه کدمربوط به جستجوی عمقی تکرارشونده است ،که این الگوریتم ازلحاظ زمانی ازمرتبه جستجوی اول سطحی است و ازلحاظ پیچیدگی حافظه ازمرتبه جستجوی اول عمق بهره میبرد. جست و جوی عمقی تکراری ، یک استراتژی کلی است . این الگوریتم با شروع از مقدار صفر به عنوان عمق محدود ، مقدارآن را به تدیج اضافه می کند مانند یک و .. تا ایکه هدفی پیدا شود .

هدف وقتی پیدا می شود که عمق محدود به d برسد ،که d عمق مربوط به عمیق ترین گره هدف است . این الگوریتم از مزایای جست و جوی عمقی و جست وجوی عرضی استفاده می کند فواید مربوط به این دو الگوریتم را با هم ترکیب می کند . این الگوریتم برای تعیین عمق محدود است که جست و جو با عمق محدود را با حدود صعودی تکرار می کند و زمانی خاتمه می یابد که جوابی پیدا شود یا جست و جو با عمق محدود مقدار failure را برگرداند که این عمل نشان می دهد جوابی وجود ندارد . \\\\\\\\\\\\\\\\\\\\\\\\\ 10 شش نوع جست و جو های ناآگاهانه جدول زیر را به تفکیک ، با چهار معیار مربوطه به اختصار شرح دهید ؟ 1(جست و جوی سطحی کامل بودن : بله / شرط : جواب بهینه در عمق d قابل دسترس باشد . فاکتور انشعاب b محدود باشد . بهینه بودن : بله / شرط : مسیر ها فاقد هزینه باشند . پیچیدگی زمانی : گره ریشه حداکثر دارای b فرزند است / هر فرزند نیز حداکثر دارای b فرزند است بنابراین در سطح دوم 2 bگره وجود دارد / با فرض اینکه جواب در عمق d باشد در بدترین حالت جواب باید در سمت راست ترین گره باشد پیچیدگی حافظه : هم مرتبه پیچیدگی زمانی است . 2(جست و جو با هزینه یکنواخت کامل بودن : بله / شرط : جواب در عمق قابل دسترس باشد . هزینه ها مقدار مثبت داشته باشند . بهینه بودن : بله / شرط : کامل باشد . پیچیدگی زمانی : فرض شود c هزینه مسیر بهینه است . فرض شود هزینه هر عمل حداقل e است . در بدترین حالت . است زمانی پیچیدگی( o(b^ce( پیچیدگی حافظه : هم مرتبه پیچیدگی زمانی است . 3(جست و جوی عمق ی کامل بودن : خی ر / شرط : مگر اینکه فضای حالت محدود باشد و حلقه تکرار وجود نداشته باشد . بهینه بودن : خیر / زیرا کامل نیست . پیچیدگی زمانی:) m^ob)است، اگر m خیلی بزرگتر از d باشد به مراتب بدتر است . در بسیاری از مسائل سریعتر از جست و جوی BF است . پیچیدگی حافظه:) +1bm(O در زمان عقبگرد حافظه آزاد می شود . 4(جست و جوی عمقی محدود در حقیقت DF با عمق محدود L است . تعیین در همه مسائل امکان پذیر نمی باشد . اگر d<L آنگاه غیر کامل است . اگر d>L آنگاه کامل اما غیر بهینه است . اگر d=L آنگاه کامل و بهینه است . پیچیدگی زمانی : ) 1b(O پیچیدگی حافظه:(bl(O 5(جست و جوی عمق ی تکراری کامل بودن : بله / شرط : حلقه تکرار وجود نداشته باشد . بهینه بودن : بله / اگر هزینه مسیر ها با هم برابر باشد .

O(b^d) : زمانی پیچیدگی پیچیدگی حافظه :(bd(O 6(جست و جوی دو طرف ه کامل بودن : بله / شرط : استفاده کردن از جست و جوی سطری بهینه بودن : بل ه / شرط : استفاده کردن از جست و جوی سطری O(b^d/2):زمانی پیچیدگی O(b^d/2):حافظه پی \\\\\\\\\\\\\\\\\\\\\\\\\ 11 جست وجوی \*A را با توجه به جدول SLD h با جست و جوی حریصانه search Greedy با رسم درختی به طور کامل توضیح داده و تفاوت ها را با دلیل ذکر کنید ؟ دراین روش گره هارا باترکیب (n(g یعنی هزینه رسیدن به گره و (n(h یعنی هزینه رسیدن ازاین گره به گره هدف ارزیابی می کند. (n(h+)n(g)=n(F یعنی (n(f هزینه برآوردشده ی ارزانترین جوار ازطریق n است.پس باید به گره ای فکرکنیم که کمترین (n(gو (n(h راداشته باشد. شناخته شده ترین جستجوی آگاهانه •ایده: از بسط گرههایی که به صرفه به نظر نمیرسند، اجتناب میکند . : f( n)= g( n) +h( n)ارزیابی تابع• (n( g• هزینه واقعی از گره شروع تا گرهn (n( h• هزینه تخمینی از گره n تا هدف (n( f• هزینه تخمینی از گره شروع تا هدف با عبور از گرهn جستجوی \*A کامل وبهینه وبهینه موثراست.مرتبه زمانی ومکانی آن نمایی است.

جستجوی حریصانه:(n( h)= n( f : گره ایی را بسط م یدهد که به هدف نزدیکتر باشد . این جستجوکامل نیست چون حلقه تکراردارد وبهینه هم نیست ومرتبه زمانی ومکانی ان( (m^b(O )است . تفاوت الگوریتم حریصانه \*Aدر(n(g یعنی هزینه واقعی است. Aجستجورابهینه وکامل میکند.جستجوی حریصامه زودتصمیم می گیرد ،امادرA مینیمم ترین گره انتخاب شده وبه آن مینیمم هزینه واقعی اعتمادمیکند. \\\\\\\\\\\\\\\\\\\\\\\\\ 12 الگوریتم زیر را شرح دهید و با توجه به جدول و شکل سوال 11 با رسم درخت جست و جو توضیح دهید ؟ این الگوریتم RBFSاست که درآن : 1(بهترین گره برگ و بهترین جانشین برای آن انتخاب شود . 2(اگر مقدار بهترین گره برگ از جانشین آن بیشتر شد، آنگاه به مسیر جانشین عقبگرد شود . 3(در حین عقبگرد، مقدار (n( f بروزرسانی شود . 4(گره جانشین بسط داده شود . RBFSجستجوی به مراتب موثرتری از \*A ID است . از تولید تعداد بسیار زیادی گره به دلیل تغییر عقیده رنج می برد . مانند \*A اگر (n( h قابل پذیرش باشد، بهینه است . پیچیدگی حافظه ( bd( o است . پیچیدگی زمانی به کیفیت تابع هیوریستیک و میزان تغییر عقیده بستگی دارد . \\\\\\\\\\\\\\\\\\\\\\\\\ 13 چند نوع تابع هیوریستیک را می توان برای پازل اعداد معرفی کرد ، با رسم شکل بررسی کنید ؟ تابع هیوریستیک قابل پذیرش1 •از طریق نسخه ساده شده از مساله(version relax 1h.هر کاشی می تواند به هرجایی منتقل شود 2h ...هر کاشی می تواند به هر خانه همسایه منتقل شود. ABSolver..هزینه راه حل برای مکعب روبیک را تخمین میزند . ابداع تابع هیوریستیک قابل پذیرش (3) •از طریق یادگیر ی از تجربه( experience experience from learning) تجربه : حل تعداد بسیار زیادی از مساله \\\\\\\\\\\\\\\\\\\\\\\\\ 14 سه راه حل جهت ابداع تابع هیوریستیک نام برده و شرح دهید ؟ 1(از طریق نسخه ساده شده از مساله 1 Hهر کاشی می تواند به هر جایی منتقل شود . 2 Hهر کاشی می تواند به هر خانه همسایه منتقل شود . ABSoloverهزینه راه حل برای مکعب روبیک را تخمین می زند . 2(از طریق نسخه کوچکتر از مساله 3(از طریق یادگیری از تجربه تجربه : حل تعداد بسیار زیادی از مساله \\\\\\\\\\\\\\\\\\\\\\\\\ 15 انواع جست و جوی محلی را نام برده و ایده هر یک را بیان کنید ؟ جست و جوی تپه نوردی ، SA، پرتو محلی ، ژنتیک الگوریتم جست و جوی محلی تپه نوردی : این الگوریتم حلقه ای است که در جهت افزایش مقدار حرکت می کند )به طرف باالی تپه ( . وقتی به قله ای رسید که هیچ همسایه ای از آن بلند تر نیست خاتمه می یابد . الگوریتم جست و جوی محلیSA :این الگوریتم نسخه ای از تپه نوردی اتفاقی است و پایین آمدن از تپه مجاز است . حرکت به طرف پایین و به آسانی در اوایل زمانبندی annealing پذیرفته شده و با گذشت طمان کمتر اتفاق می افتد . الگوریتم جست و جوی پرتو محلی : نگهدار ی فقط یک گره در حافظه ، واکنش افراطی نسبت به مسئله محدودیت حافظه است . این الگوریتم به جای بک حالت ، kحالت را نگهداری می کند . این الگوریتم با kحالت که به طور تصادفی تولید شدند ، شرو ع می کند . در هر مرحله تمام پسین های همه حالت ها تولید می شوند . اگر یکی از آن ها هدف بود ، الگوریتم متوقف می شود ؛ وگرنه بهترین پسین را انتخاب و عمل را تکرار می کند . الگوریتم جست و جوی محلی ژنتیک : این الگوریتم شکلی از جست و جوی پرتو اتفاقی است که در آن ، حالت های پسین از طریق ترکیب دو حالت والد تولید می شوند . در مقایسه با انتخاب طبیعی ، مثل جست و جوی پرتو اتفاقی است ، با این تفاوت که اینجا با تولید مثل جنسی سروکار داریم نه غیر جنسی .

این الگوریتم همانند جست و جوی پرتو محلی ، با مجموعه ای از k حالت که به طور تصادفی تولید شدند شروع می کند که به آن جعیت گفته می شود \\\\\\\\\\\\\\\\\\\\\\\\\ 16 الگوریتم زیر را شرح داده و انواع آن را نام برده و بررسی کنید ؟ الگوریتم باال مربوط به الگوریتم جست و جوی محلی تپه نوردی می باشد . این الگوریتم حلقه ای است که در جهت افزایش مقدار حرکت می کند ) به طرف باالی تپه( . وقتی به قله ای رسید که هیچ همسایه ای از آن بلند تر نیست خاتمه می یابد . در این الگوریتم درخت جست و جو را نگهداری نمی کند . لذا ساختمان داده گره فعلی فقط باید حالت و مقدار تابع هدف را نگهداری کند . تپه نوردی به همسایه های حالت فعلی نگاه می کند . مثل تالش برای یافتن قله کوه اورست در مه گرفتگی غلیظ ، در حالی که دچار فراموشی هستید . تپه نورد ی گاهی جست و جوی محلی حریصانه نام دارد زیرا بدون اینکه قبال فکر کند به کجا برود ، حالت همسایه خوبی را انتخاب می کند . تپه نوردی معموال به سرعت به جواب پیش می رود ، زیرا به راحتی می تواند حالت بد را بهبود ببخشد .

انواع تپه نوردی ؛ تپه نوردی غیر قطعی : در ب ین حرکت های رو به باال یکی به صورت تصادفی انتخاب می شود . البته احتمال انتخاب با شیب متناسب است . تپه نوردی با انتخاب اولین گزینه : گره ها تا حصول یک گره بهتر بسط داده می شوند . تپه نوردی تصادفی : از حالت شروع مجدد تصادفی تا حصول جواب مجددا شروع خواهد نمود.