کویز شماره ۲

بهاره باقری

سکشن چهارشنبه ۱۵:۳۰ – ۱۸

سوال ۱ :

۱- فرمول بندی مساله :

در این مرحله، ابتدا مساله را به صورت دقیق تعریف می کنیم. در مثال شهرهای رومانی، فرض کنید می‌خواهیم کوتاه ترین مسیر از شهر "آراد" به "بخارست" را پیدا کنیم. مساله این است که بهترین (کوتاه‌ترین) مسیر را برای رسیدن از مبدا به مقصد پیدا کنیم.

۲- تولید راه حل ها :

در این مرحله، راه حل های مختلف را با استفاده از استراتژی های مختلف جستجو (مثل جستجوی عرضی، عمقی، یا اکتشافی) بررسی می‌کنیم. هر مسیر ممکن، یک راه حل بالقوه برای رسیدن از آراد به بخارست است و باید بررسی شود که آیا به مقصد می‌رسد یا خیر.

۳- انتخاب بهترین راه حل :

بعد از تولید راه حل ها، راه حل های به دست آمده را ارزیابی کرده و بهترین راه حل را انتخاب می‌کنیم. در این مثال، انتخاب بهترین مسیر به این معنی است که مسیری را پیدا کنیم که کمترین هزینه (مسافت یا زمان) را داشته باشد و در نهایت به بخارست برسد.

٤- اجرای راه حل :

پس از انتخاب بهترین مسیر، آن را اجرا می کنیم؛ یعنی با حرکت از آراد به شهرهای مختلف، در نهایت به بخارست می‌رسیم.

.....

سوال ۲ :

۱- **مساله با حالت ایستا** :در این نوع مسائل، شرایط و قوانین مساله در طول زمان تغییر نمی‌کنند و ثابت هستند. به عنوان مثال، **مساله شطرنج** در حالت ایستا در نظر گرفته می‌شود، زیرا قوانین و صفحه شطرنج در طول بازی تغییر نمی‌کنند. بازیکنان فقط باید بر اساس موقعیت موجود تصمیم بگیرند.

۲- **مساله دینامیک** :در مسائل دینامیک، شرایط مساله در طول زمان تغییر می‌کنند و باید به سرعت واکنش نشان داد. برای مثال، **مساله رانندگی در ترافیک** یک مساله دینامیک است؛ چون شرایط جاده، ترافیک و تصمیم‌های دیگر رانندگان ممکن است به سرعت تغییر کنند و راننده باید به این تغییرات پاسخ دهد.

۳- **مساله با اطلاعات کامل** :در این نوع مساله، همه اطلاعات لازم برای حل مساله در دسترس است. مثالی از این نوع **مساله بازی شطرنج** است، چون وضعیت صفحه شطرنج کاملاً قابل مشاهده است و هیچ اطلاعات پنهانی وجود ندارد.

٤- **مساله با اطلاعات ناقص** :در این نوع مساله، اطلاعات کامل درباره شرایط یا وضعیت‌ها در دسترس نیست. به عنوان مثال، **مساله جستجو و نجات** در یک منطقه ناشناخته را در نظر بگیرید؛ نجات‌دهنده ممکن است اطلاعات کامل درباره وضعیت منطقه و خطرات نداشته باشد.

٥- **مساله قابل تقسیم** :در این نوع مسائل، مساله اصلی را می‌توان به زیرمساله‌های کوچکتر تقسیم کرد و آن‌ها را به طور جداگانه حل کرد. به عنوان مثال، **مساله مرتب‌سازی** یک آرایه از اعداد قابل تقسیم است، زیرا می‌توان آن را به دو زیرآرایه تقسیم کرده و هرکدام را جداگانه مرتب کرد.

٦- **مساله تک‌مرحله‌ای** :این نوع مسائل فقط یک گام برای رسیدن به پاسخ نیاز دارند. مثالی از این نوع، **مساله ریاضی ساده** است، مثل محاسبه یک معادله، که در یک مرحله حل می‌شود.

۷- مساله چندمرحله‌ای :در این نوع مساله، نیاز به چندین گام یا مرحله برای رسیدن به پاسخ داریم. برای مثال، **مساله مسیریابی** که شامل حرکت از یک نقطه به نقطه دیگر در نقشه است، نیاز به چندین گام دارد و هر گام می‌تواند به انتخاب مسیر بهتر کمک کند.

.....

سوال ۳ :

۱- روش مبتنی بر وضعیت :

- **حالت اولیه** :صفحه شطرنج خالی است و هیچ وزیری روی آن قرار ندارد.
- **عملگرها** :اضافه کردن یک وزیر به هر سطر خالی صفحه شطرنج. در این روش، هر بار یک وزیر را در یک سطر جدید اضافه می‌کنیم، با شرط اینکه وزیر جدید در هیچ ستونی یا قطری با وزیران قبلی تهدیدکننده نباشد.
- **هدف** :حالتی که ۸ وزیر در صفحه شطرنج قرار گرفته‌اند و هیچ کدام از آن‌ها نمی‌توانند دیگری را تهدید کنند. یعنی هر وزیر در یک سطر، یک ستون و یک قطر متفاوت قرار بگیرد.

مثال :فرض کنید وزیر اول را در خانه (۱،۱) قرار می‌دهیم. برای قرار دادن وزیر دوم، باید سطر دوم را انتخاب کنیم، اما خانه‌هایی که در همان ستون یا قطر وزیر اول قرار دارند، غیرقابل استفاده هستند. این روند ادامه می‌یابد تا همه وزیران با رعایت شرایط روی صفحه قرار گیرند.

۲- روش محدودیت‌محور :

- **متغیرها** :هر وزیر در یک سطر قرار می‌گیرد و موقعیت آن در ستون مشخص می‌شود. بنابراین، ۸ متغیر برای ۸ سطر داریم که هر کدام یک مقدار از ۱ تا ۸ (ستون‌های صفحه) می‌گیرند.
- **محدودیت‌ها** :محدودیت‌ها شامل شرایطی هستند که دو وزیر نمی‌توانند در یک سطر، یک ستون یا یک قطر قرار بگیرند.
  - هیچ دو متغیری نمی‌توانند مقدار یکسانی داشته باشند (دو وزیر نمی‌توانند در یک ستون باشند).
  - هیچ دو متغیری نباید در موقعیتی باشند که فاصله مطلق سطر و ستون آن‌ها برابر باشد (عدم تهدید قطری).
- **هدف** :یافتن یک تخصیص برای متغیرها به‌طوری که همه محدودیت‌ها رعایت شوند.

مثال :با تعیین موقعیت وزیر اول، محدودیت‌هایی برای انتخاب‌های بعدی ایجاد می‌شود. فرض کنید وزیر اول در ستون ۱ قرار دارد، وزیر دوم نمی‌تواند در ستون ۱ یا در موقعیت‌های قطری نسبت به وزیر اول قرار گیرد، و این روند برای هر وزیر ادامه می‌یابد تا یک تخصیص معتبر پیدا شود.

.....

سوال ٤ :

۱- جستجوی عمق اول :

در جستجوی عمق اول، ابتدا به عمق درخت حرکت می‌کنیم تا به پایین‌ترین گره برسیم. سپس اگر به گره هدف نرسیده باشیم، به عقب بازمی‌گردیم و شاخه‌های دیگر را بررسی می‌کنیم.

مثال: فرض کنید می‌خواهیم مسیر یک درخت ساده با گره‌های A (ریشه)، B، C، D و E را پیدا کنیم که ساختار درخت به این صورت است:

گره A به گره‌های B و C متصل است.

گره B به گره‌های D و E متصل است.

با جستجوی عمق اول، مسیر به این ترتیب خواهد بود: A-> B-> D (تا انتها می‌رویم)، سپس عقب‌گرد می‌کنیم و E را بررسی می‌کنیم. اگر هدف پیدا نشود، سپس به شاخه C می‌رویم.

۲- جستجوی عرض اول :

در جستجوی عرض اول، به جای حرکت به عمق درخت، ابتدا تمام گره‌های مجاور سطح فعلی را بررسی می‌کنیم و سپس به سطوح بعدی می‌رویم.

مثال: در همان درخت با گره‌های A، B، C، D و E:

ابتدا گره A بررسی می‌شود.

سپس به گره‌های B و C (که در سطح بعدی هستند) می‌رویم.

در نهایت به گره‌های D و E که به سطح B متصل هستند، حرکت می‌کنیم.

هر دو روش کاربردهای خاص خود را دارند:

جستجوی عمق اول برای مسائل با محدودیت حافظه مناسب‌تر است، اما ممکن است به مسیرهای بی‌انتها برخورد کند.

جستجوی عرض اول برای یافتن کوتاه‌ترین مسیر مناسب است، اما به حافظه بیشتری نیاز دارد.

.....

سوال ۵ : فضای حالت یا **Fringe** مجموعه‌ای از تمام وضعیت‌ها (حالات) یا مسیرهای ممکن است که در طول جستجو از وضعیت اولیه تا وضعیت هدف ایجاد می‌شود. فضای حالت، تمام حالت‌هایی که ممکن است در فرایند جستجو ایجاد شوند، به همراه ارتباط‌ها و مسیرهای ممکن بین آن‌ها را شامل می‌شود.

در فرآیند جستجو، فضای حالت به عنوان محدوده‌ای برای گسترش مسیرهای جستجو به کار می‌رود. هر گره در فضای حالت، یک حالت از مساله را نشان می‌دهد که به طور معمول شامل موقعیت فعلی، وضعیت فعلی و راه‌های ممکن برای حرکت به سمت حالت بعدی است.

.....

سوال ۶ :جستجوی ناآگاهانه نوعی روش جستجو است که در آن هیچ اطلاعاتی در مورد فاصله تا هدف یا هزینه مسیر در دسترس نیست. این روش‌ها فقط از ساختار مساله و اطلاعات پایه‌ای آن برای پیدا کردن راه‌حل استفاده می‌کنند و هیچ دانش خاصی در مورد محل قرارگیری هدف ندارند. به همین دلیل، این جستجوها تمام حالات ممکن را با الگویی مشخص بررسی می‌کنند و به دنبال رسیدن به هدف هستند.

انواع جستجوی ناآگاهانه:

۱- جستجوی عرض اول :

در این روش، جستجو از ریشه آغاز می‌شود و به ترتیب هر سطح از درخت یا گراف به صورت عرضی بررسی می‌شود تا به گره هدف برسیم. این روش برای پیدا کردن مسیرهای کوتاه‌تر مناسب است اما به حافظه زیادی نیاز دارد.

۲- جستجوی عمق اول :

در جستجوی عمق اول، ابتدا به عمق درخت یا گراف می‌رویم تا به پایین‌ترین گره برسیم. اگر به گره هدف نرسیده باشیم، به عقب برمی‌گردیم و شاخه‌های دیگر را بررسی می‌کنیم. این روش به حافظه کمتری نیاز دارد اما ممکن است در مسیرهای بی‌انتها گیر کند.

۳- جستجوی عمق محدود :

این روش مشابه جستجوی عمق اول است اما با یک حد عمق مشخص. یعنی پس از رسیدن به عمق تعیین شده، دیگر به عمق بیشتری نمی‌رود و در صورت عدم پیدا کردن هدف، جستجو را متوقف می‌کند. این روش مشکل گیر کردن در مسیرهای بی‌انتها را حل می‌کند.

۴- جستجوی عمق‌افزا :

در این روش، جستجو با یک حد عمق کم شروع می‌شود و به تدریج حد عمق افزایش می‌یابد تا به هدف برسیم. این روش مزایای جستجوی عمق اول و عرض اول را ترکیب می‌کند و برای پیدا کردن راه‌حل با کمترین هزینه در فضای جستجوی بزرگ مناسب است.

۵- جستجوی هزینه یکنواخت :

در این روش، جستجو به سمت گره‌هایی که هزینه مسیر کمتری دارند، انجام می‌شود. به عبارت دیگر، ابتدا مسیری با کمترین هزینه را انتخاب کرده و به سمت آن حرکت می‌کند. این روش برای یافتن کوتاه‌ترین مسیر در گراف‌های وزن‌دار مناسب است.

.....

سوال ۷ : الگوریتمی که از لحاظ پیچیدگی زمانی از مرتبه جستجوی اول سطح و از لحاظ پیچیدگی حافظه از مرتبه جستجوی اول عمق باشد، الگوریتم جستجوی عمق‌افزا نام دارد. این الگوریتم مزایای هر دو روش جستجوی عرض اول و عمق اول را ترکیب می‌کند.

شرح الگوریتم جستجوی عمق‌افزا :

جستجوی عمق افزا در واقع ترکیبی از عمق اول و عرض اول است. در این الگوریتم، جستجو در چندین مرحله انجام می‌شود، به این صورت که در هر مرحله، عمق مشخصی تعیین شده و الگوریتم جستجوی عمق اول با محدودیت عمق اجرا می‌شود. اگر در این مرحله گره هدف پیدا نشود، عمق افزایش می‌یابد و جستجو دوباره انجام می‌شود.

مزایا و ویژگی‌ها:

۱. **پیچیدگی زمانی مشابه BFS :**این الگوریتم تمام حالات را مانند BFS بررسی می‌کند، بنابراین از لحاظ پیچیدگی زمانی مشابه BFSعمل می‌کند و در نهایت، کوتاه‌ترین مسیر را پیدا می‌کند.

۲. **پیچیدگی حافظه مشابه DFS :** برخلاف BFS که به حافظه زیادی نیاز دارد، IDDFSتنها به حافظه‌ای برابر با عمق فعلی نیاز دارد) مانند .(DFS بنابراین، پیچیدگی حافظه آن بهینه و از مرتبه DFS است.

.....

سوال ۸ : در جستجوهای ناآگاهانه، هر الگوریتم عملکرد متفاوتی از نظر **کامل بودن، بهینگی، پیچیدگی زمانی** و **پیچیدگی فضایی** دارد. در زیر کارایی هر یک از انواع جستجوهای ناآگاهانه بر اساس این چهار پارامتر آورده شده است:

۱. **جستجوی عرض اول :**

- **کامل بودن** :بله، اگر عمق هدف محدود و فضای حالت محدود باشد، BFSکامل است و در نهایت گره هدف را پیدا می‌کند.
- **بهینگی** :بله، اگر هزینه هر مرحله برابر باشد، این الگوریتم کوتاه‌ترین مسیر را پیدا می‌کند.
- **پیچیدگی زمانی :** $O(b^d)$ ، که در آن b تعداد فرزندان هر گره و d عمق هدف است.
- **پیچیدگی فضایی** : $O(b^d)$ ، چون نیاز به نگهداری همه گره‌های سطح فعلی دارد و حافظه زیادی مصرف می‌کند.

۲. **جستجوی عمق اول :**

- **کامل بودن** :خیر، اگر درخت بی‌انتها باشد، ممکن است DFS هیچگاه به پایان نرسد.
- **بهینگی** :خیر، این الگوریتم بهینه نیست و ممکن است به اولین پاسخ برسد، نه لزوماً کوتاه‌ترین مسیر.
- **پیچیدگی زمانی :** $O(b^m)$ ، که در آن m بیشترین عمق درخت است. این روش در بدترین حالت زمان زیادی مصرف می‌کند.
- **پیچیدگی فضایی :** $O(b^m)$ ، که نیاز به حافظه بسیار کمتری نسبت به BFS دارد، زیرا فقط مسیر جاری در حافظه نگه داشته می‌شود.

۳. **جستجوی عمق محدود**:

- **کامل بودن** :اگر حد عمق به اندازه کافی بزرگ باشد که به هدف برسد، بله.
- **بهینگی** :خیر، ممکن است کوتاه‌ترین مسیر را پیدا نکند.
- **پیچیدگی زمانی :** $O(b^l)$ ، که در آن l همان حد عمق تعیین شده است.
- **پیچیدگی فضایی :** $O(b^l)$، که نسبت به DFS و BFS بهینه‌تر است.

۴. **جستجوی عمق‌افزا :**

- **کامل بودن** :بله، این الگوریتم کامل است و در نهایت به هدف می‌رسد.
- **بهینگی** :بله، اگر هزینه مراحل برابر باشد، کوتاه‌ترین مسیر را پیدا می‌کند.
- **پیچیدگی زمانی :** $O(b^d)$ ، مشابهBFS ، چون در هر عمق جستجو را تکرار می‌کند.
- **پیچیدگی فضایی :** $O(b^d)$ که بسیار کمتر از BFS است و مشابه DFS عمل می‌کند.

۵. **جستجوی هزینه یکنواخت :**

- **کامل بودن** :بله، UCSکامل است.
- **بهینگی** :بله، UCSبهینه است و همیشه مسیر با کمترین هزینه را پیدا می‌کند.
- **پیچیدگی زمانی :** $O(b^{\lceil C^*/\epsilon \rceil})$ ، که در آن C^* هزینه مسیر بهینه و ϵ کمترین اختلاف بین هزینه‌ها است.
- **پیچیدگی فضایی :** $O(b^{\lceil C^*/\epsilon \rceil})$ ، به دلیل نیاز به نگهداری مسیرهای ممکن در حافظه.