حدیث قائدشرفی/کوییز فصل 3/چهارشنبه سکشن 15:45

- **تعریف مسئله:** در این مرحله، مسئله‌ای که قرار است حل شود، به‌طور دقیق تعریف می‌شود. در مثال 1. "مسیریابی در شهرهای رومانی"، مسئله این است که کوتاه‌ترین مسیر بین دو شهر (مثلاً از بخارست به اورادئا) را پیدا کنیم.
- **طراحی راه‌حل:** الگوریتم مناسب برای حل مسئله انتخاب می‌شود. برای مثال، از الگوریتم‌های مسیریابی مانند **الگوریتم دایکسترا** یا **جستجوی A*** استفاده می‌کنیم که بهترین مسیر با کمترین هزینه (مثلاً مسافت) را پیدا می‌کند.
- **پیاده‌سازی راه‌حل:** الگوریتم انتخاب‌شده پیاده‌سازی می‌شود. مثلاً، گرافی از شهرهای رومانی که با یال‌ها و گره‌ها نشان داده شده و هر یال نمایانگر مسافت بین دو شهر است، در برنامه‌ای برای الگوریتم جستجوی A* قرار می‌گیرد.
- **ارزیابی و بهینه‌سازی:** نتایج بررسی و بهینه‌سازی می‌شوند. پس از یافتن مسیر، طول و کارایی آن ارزیابی می‌شود و در صورت نیاز راه‌حل بهینه‌تری ارائه می‌شود، تا مطمئن شویم مسیر بهینه برای کاربر مناسب است.

2. مسایل تك حالته: State Single) (قطعی وکاملاً قابل مشاهده: در این مسائل، عمل ی کسان همیشه نت یجهای یکسان خواهد داشت. مثل حل مکعب روبیک قطعی وبخشی قابل مشاهده: مسایل غیر قابل این :Contingency) (غیر قطعی وبخشی قابل مشاهده: مسایل احتمالی • :Conformant/Sensorless) (دریافت مسائل شامل عدم قطعی ت هستند، یعنی اعمال ممکن است به نتایج مختلفی منجر شوند. مثال: ربات جاروبرقی • فضای این نوع مسائل حالتی دارند که همه اطالعات مورد :Online/Exploration) (حالت ناشناخته: مسایل اکتشافی یا برخط نیاز در هر لحظه در دسترس است، ی عنی عامل تمام جزئیات الزم برای تصمی م گیری را دارد. مثال: بازی شطرنج

3. 1 فرموله سازی افزایشی: • حاالت: جایگشت های مختلف چینش • حالت شروع: صفحه خالی • اعمال: اضافه نمودن هر وزیر در یک ستون • ازمون هدف: 8 وزیر بر روی صفحه شطرنج .2 فرموله سازی کامل: • حاالت: جایگشت های مختلف چینش • حلت شروع: هر 8 وزیر بر روی صفحه • اعمال: جا به جا نمودن وزیر ها در صفحه • ازمون هدف: عدم تهدید وزیر ها

4. مثال شهر اراد: جستجوی فضای حالت با تولید یک درخت ریشه=حالت شروع)شهر اراد(گره ها)شهر هایی که دارای بسط هستند(برگ ها)شهر هایی که بسط ندارند(از طریق توابع جانشینی تولید می شوند در حالت کلی جستجو منجر به تولید گراف می شود

5. فضای حالت مجموعهای از تمام حالتهای ممکن برای مسئله است که شامل همه وضعیتهایی است که عامل می تواند در طول حل مسئله به آنها برسد. در فضای حالت، هر گره یک حالت است و هر لبه بین گره ها یک عمل یا تغییری است که مجموعه ای از گره ها یی است که تا این لحظه Fringe می توان آن را بر روی حالت انجام داد تا به حالت بعدی برس یم . گره Fringe بررسی نشده اند اما آماده بررسی در مرحله بعدی هستند. در هر الگوری تم جستجو، گره های موجود در هایی هستند که به فضای جستجوی عامل اضافه شده اند و منتظرند تا بررسی شوند.

6. جستجوی ناآگاهانه فقط از اطالعات موجود در مسئله استفاده می نماید.)جستجوی کورکورانه(انواع جستجو : 1(جستجوی سطحی 2(جستجوی هزینه یکنواخت 3(جستجوی عمقی 4(جستجوی عمقی محدود 5(جستجوی عمقی تکرار شونده 6(جستجوی دو طرفه

7. **الگوریتم جستجوی عمق‌افزایشی (Iterative Deepening Search - IDS)** عمل می‌کند، که از لحاظ پیچیدگی زمانی مشابه **جستجوی عرض اول (BFS)** و از لحاظ پیچیدگی حافظه مشابه **جستجوی عمق اول (DFS)** است. این الگوریتم ویژگی‌های هر دو روش را ترکیب کرده و در بسیاری از موارد مزایای هر دو را به‌طور همزمان ارائه می‌دهد.

شرح الگوریتم جستجوی عمق‌افزایشی (IDS)

در الگوریتم جستجوی عمق‌افزایشی، جستجو در چندین مرحله انجام می‌شود، که در هر مرحله عمق جستجو افزایش می‌یابد. به این صورت که ابتدا جستجو را با عمق ۱ انجام می‌دهد، سپس عمق را به ۲ افزایش می‌دهد و جستجوی جدیدی را تا این عمق انجام می‌دهد و این روند را تا رسیدن به هدف ادامه می‌دهد.

8. جستجوی سطح اول (BFS :) از لحاظ پیچیدگی زمانی و فضایی به صورت نمایی رشد می کند. • جستجوی عمق اول (DFS :) پیچیدگی زمانی ان نمایی ولی پیچیدگی فضایی ان خطی است. این روش کامل و بهینه نیس ت. این روش کامل و بهینه است)در صورتی که هزینه یال ها یکسان باشد(.