

* اگر به الگوریتمی هیچ اطلاعاتی به جز تعریف مسئله داده نشود، آن الگوریتم را ناآگاه می‌گوییم و تنها قادر

به تولید پسین‌ها و تشخیص حالت هدف از غیر هدفمند

□ Breadth-first search: جستجوی اول سطح

□ Uniform-cost search: جستجوی هزینه یکنواخت

□ Depth-first search: جستجوی اول عمیق

□ Depth-limited search: جستجوی عمیق محدود

■ Iterative deepening search: جستجوی عمیق تکراری

□ Bidirectional search: جستجوی دو طرفه

جستجوی عمیق کننده تکراری:

قسمت دشوار جستجوی عمیق محدود شده انتخاب یک محدودره خوب است

اگر محدودره عمیق بهتری را پیدا کنیم این محدودره ما را به سوی جستجوی کارآمدتری می‌رساند
اما برای بیشتر مسائل محدودره عمیق مناسب را تا زمانی که مسئله حل نشده است نمی‌شناسیم
جستجوی عمیق کننده تکراری استراتژی است که نظریه انتخاب بهترین محدودره عمیق توسط
امتحان تمام محدودره‌های ممکن را یادآوری می‌کند

و برایابی جستجوی اول سطح و اول عمیق را یکجا دارد

این جستجو مانند جستجوی سطحی کامل روبه‌پیش است؛ اما فقط مزیت درخواست حافظه اندک را از جستجوی عمیق دارد.

در جستجوی عمیق کننده تکراری، اگر فضای سطحی یک بار بسط داده می‌شود، آن‌هایی که یک سطح بالاتر قرار دارند دوبار بسط داده می‌شوند و آن آخر تا به ریشه درخت جستجو برسیم، که $d+1$ بار بسط داده می‌شوند
بنابراین مجموع دفعات بسط در این جستجو عبارت است از: $3b^{d-2} + \dots + (d-1)b^2 + (d)b^1 + (d+1)b^0$
در حالت کلی عمیق کننده تکراری، ارزش جستجوی بهتری است؛ زمانی که فضای جستجوی بزرگی وجود دارد و عمیق راه حل
ذیرمجهول است 1- کامل است 2- پیچیدگی زمانی $O(b^d)$ 3- پیچیدگی فضایی $O(bd)$
4- اگر هزینه مسیر را در نظر بگیریم بهینه است

چگونه یک جابجایی را می‌توان با استفاده از یک سنسور ساده تشخیص دیوار در الگوریتم جستجوی

فاکتافانه عمیق شونده تکراری (Iterative Deepening Depth-First search - IDDFS) خروجی یک اتاق را پیدا کند مراحل زیر را در نظر می‌گیریم.

1- مقدار دهی اولیه: جابجایی را با یک با Limit اولیه برابر با 1 سانتی‌متر شروع می‌کند این به این معنی است که در اولین تلاش، جابجایی فقط 1cm از نقطه شروع خود حرکت می‌کند

2- تشخیص دیوار: سنسور جابجایی 0 یا 1 را می‌گیرد؛ 0 که صفر به معنی عدم وجود دیوار و 1 که به معنی وجود دیوار است

3- جستجوی IDDFS: جابجایی از الگوریتم IDDFS برای جستجوی فضای حالت استفاده می‌کند

در هر تکرار - Limit به اندازه 1 سانتی‌متر افزایش می‌یابد و جابجایی تا آن فاصله جستجو می‌کند اگر جابجایی در Limit مغلی خروجی را نیابد به نقطه شروع بازگشته و Limit را 1cm افزایش می‌دهد و جستجو را از سر

4- یافتن خروجی: این فرآیند تا زمانی که جابجایی خروجی اتاق را پیدا کند یا تمام فضای قابل دسترس را جستجو کند ادامه می‌یابد

این الگوریتم به جابجایی امکان می‌دهد تا با استفاده از یک رویکرد جستجوی تکراری و گسترده به تدریج فضای بیشتری از اتاق را پوشش دهد و در نهایت خروجی را پیدا کند این روش می‌تواند موثر باشد زیرا جابجایی را قادر می‌سازد تا با حداقل اطلاعات اولیه بهینه‌ترین مسیر را برای پوشش دادن کل اتاق و یافتن خروجی شناسایی کند.

* باسک از یک متغیر مثل zero counter تعداد دفعاتی که سنسور خروجی صفری در بین دور حرکت

با خروجی یک شمارنده پیش‌اگر این شمارنده به دور رسد، آن را به عنوان خروجی در نظر می‌گیریم