

جستجو و جستجوی ناآگاهانه

پرسش: عامل‌های حل مساله نوعی عامل هدف گرا هستند. چرا؟

چند مثال از مساله

- Solving a crossword puzzle
- Select a good chess move
- Building a spaceship to go to Mars
- Building a successful career
- How to get an "A" in this course

تعریف مساله: مجموعه‌ای از اطلاعات؛ مجموعه‌ای از عملگرهاست که با آنها می‌توان مساله را حل کرد.

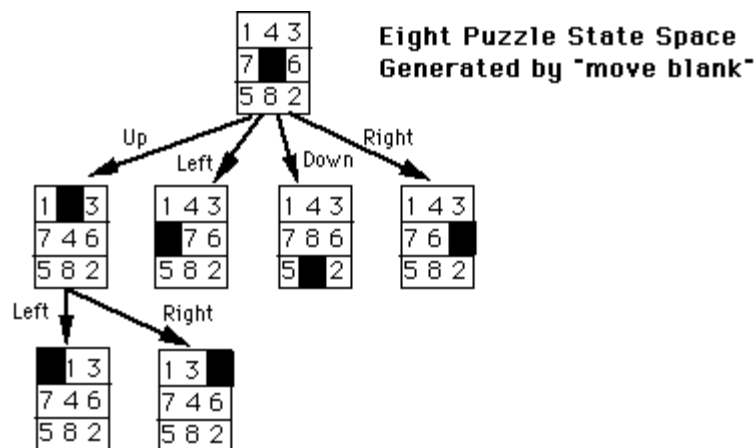
پرسش: فضای مساله در درس طراحی الگوریتم‌ها، چگونه تعریف می‌شود؟

نتیجه اینکه، حل مساله، جستجو در فضای مساله است.

درخت جستجو: فرایند جستجویی که منجر به ایجاد درخت جستجو، می‌شود. این درخت شامل ریشه، گره‌های میانی و گره‌های برگ است.

مولفه‌های اصلی در توصیف فضای مساله: ۱- وضعیت آغازین ۲- عملگرها ۳- وضعیت هدف

مثال: معمای هشت



شکل ۲-۱: معمای هشت

ساختمان داده برای درخت جستجو

- State
- Parent node
- The operator was applied to generate the node
- Depth of the node
- Path cost of the node
- Fringe: The collection of nodes that are waiting to be expanded (queue)

زندگی یعنی جستجوی دائم. (لامارتین)

حل مساله با استفاده از جستجو

پرسش: در درس طراحی الگوریتم‌ها، چه موقعی استفاده از روش‌های جستجو (جستجوی فضای حالت مساله)، نظیر روش پسگرد مطرح می‌شود؟

استراتژی جستجو

چگونگی انتخاب گره بعدی برای گسترش درخت جستجو را استراتژی جستجو گویند. این انتخاب از مجموعه گره‌ها (عناصر) صورت می‌گیرد که می‌تواند این مجموعه عناصر، در ساختمان داده‌ای نظیر یک لیست یا صف ذخیره شود و با توجه به گونه‌های مختلف آن، الگوریتم‌های جستجو مختلفی را ایجاد شود.

معیارهای مقایسه در استراتژی جستجو

- کامل بودن (Completeness)
 - پیچیدگی زمانی (Time Complexity)
 - پیچیدگی مکانی (Space Complexity)
 - بهینگی (Optimality)
- روشهای جستجو به دو دسته کلی زیر تقسیم می‌شوند:

۱- جستجوی ناآگاهانه

۲- جستجوی آگاهانه

جستجوی ناآگاهانه: جستجویی است که هیچ‌گونه آگاهی از تعداد مراحل یا هزینه مسیر از حالت جاری تا حالت هدف وجود ندارد. تنها قادر به تشخیص حالت هدف از حالت غیرهدف هستند. این روش جستجو خود شامل شش استراتژی است:

- جستجوی سطحی یا اول پهنا (Breadth-first Search)
- جستجوی عمقی یا اول عمق (Depth-first Search)
- جستجوی سازگار با هزینه (Uniform Cost Search)
- جستجوی عمقی محدود شده (Depth-limited Search)
- جستجوی عمیق کننده تکراری (Iterative Deepening Search)
- جستجوی دوطرفه (Bidirectional Search)

تعریف چند اصطلاح

d = عمق درخت جستجو

m = حداکثر عمق درخت جستجو

b = ضریب انشعاب

۱- جستجوی سطحی

ساختمان داده صف و FIFO (first-in, first-out)

کامل است. بهینه است با شرط اینکه هزینه همه عملگرها یا وزن یال‌ها در درخت جستجو یکسان باشد. اگر نباشد؟

پیچیدگی زمانی $O(b^d)$ و پیچیدگی مکانی $O(b^d)$

مثال: به شکل ۲-۲ و شکل ۴-۲ مراجعه کنید.

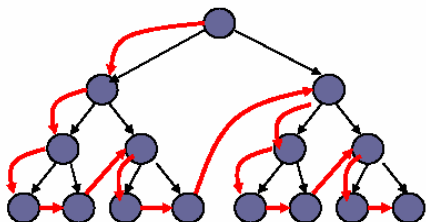
۲- جستجوی عمقی

LIFO (last-in, first-out) ساختمان داده پشته و

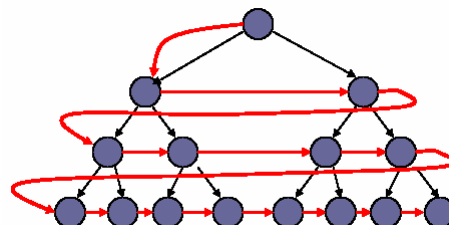
کامل نیست؛ چون ممکن است جستجو در شاخه‌ای از درخت، پایان نپذیرد. بهینه هم نیست. چرا؟

پیچیدگی زمانی $O(b^d)$ و پیچیدگی مکانی $O(bd)$

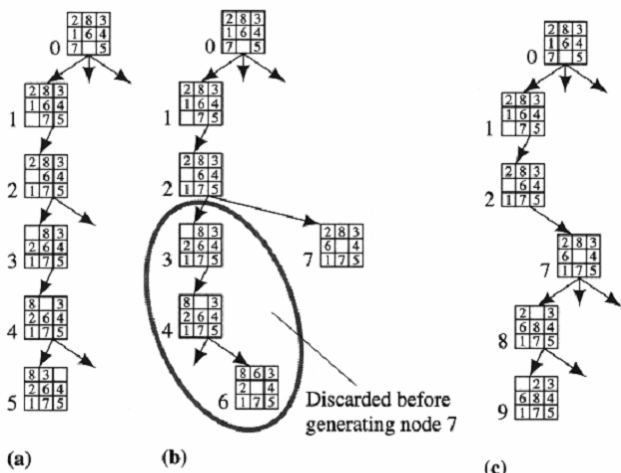
مثال: به شکل ۲-۳ و شکل ۲-۵ مراجعه کنید.



شکل ۲-۳: جستجوی عمقی

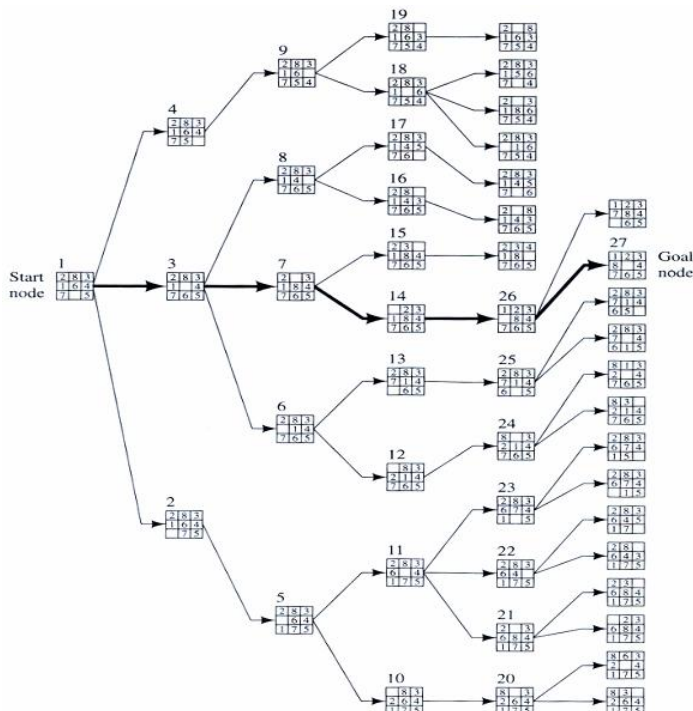


شکل ۲-۴: جستجوی سطحی



Depth-First Search

شکل ۲-۵: جستجوی عمقی برای معمای هشت



شکل ۲-۴: جستجوی سطحی برای معمای هشت

۳- جستجوی سازگار با هزینه

از تابع هزینه $g(n)$ استفاده می کند که:

$g(n)$ = path cost from the initial state to n , Expand the leaf node with the minimum first.

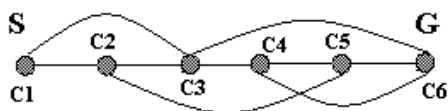
کامل است با شرط منفی نبودن هزینه عملگرها ($g(n) \leq \text{successor}(n)$) و بهینه است؛ مسیری با کمترین طول را برمی گرداند.

پیچیدگی زمانی $O(b^d)$ و پیچیدگی مکانی $O(bd)$

مثال: به کتاب مراجعه کنید.

۴- جستجوی عمقی محدود شده

مثال: به شکل های ۲-۶ و ۲-۷ مراجعه کنید.



شکل ۲-۶: مساله مسیر یابی

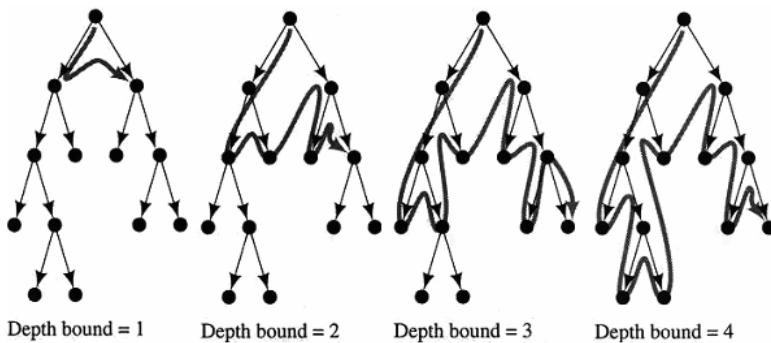
۵- جستجوی عمقی تکرارشونده یا جستجوی عمقی گام به گام

Depth-First Iterative Deepening (DFID) Or Iterative Deepening Search (IDS)

کامل است. بهینه است اگر هزینه تمام عملگرها یکسان باشد.

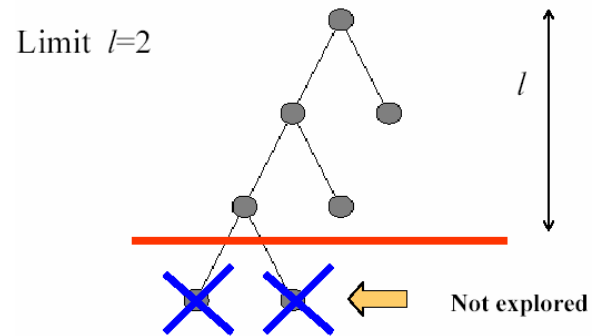
پوشش: اگر هزینه عملگرها یکسان نباشد، روش IDS چه راه حلی پیدا می کند؟ برای بهینگی آن چه تغییری باید در روشی IDS اعمال کرد؟ پیچیدگی زمانی آن کمی بدتر از جستجوی عمقی است. مرتبه پیچیدگی مکانی آن، مشابه جستجوی عمقی، خطی است.

مثال: به شکل ۸-۲ مراجعه کنید.

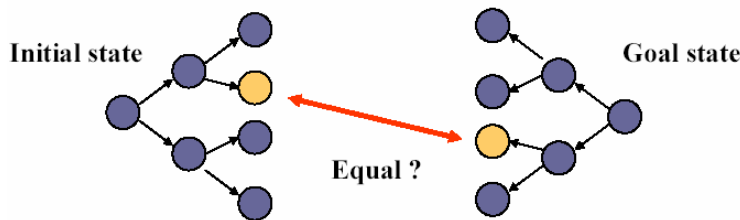


Stages in Iterative-Deepening Search

شکل ۸-۲: جستجوی عمقی تکرار شونده



شکل ۷-۲: جستجوی عمقی محدود شده



شکل ۹-۲: جستجوی دو سوپه

۶- جستجوی دو سوپه

مثال: به شکل ۹-۲ مراجعه کنید.

مقایسه استراتژی های جستجوی ناآگاهانه

Criterion	Breadth-First	Uniform-Cost	Depth-First	Depth-Limited	Iterative Deepening	Bidirectional (if applicable)
Time	b^d	b^d	b^m	b^l	b^d	$b^{d/2}$
Space	b^d	b^d	bm	bl	bd	$b^{d/2}$
Optimal?	Yes	Yes	No	No	Yes	Yes
Complete?	Yes	Yes	No	Yes, if $l \geq d$	Yes	Yes

جدول ۱-۲: مقایسه روش های جستجوی ناآگاهانه

تمرین ها

تمرین ۱-۲: برای مساله معمای هشت، کشیش ها و آدمخوارها و مساله تنگ آب، ساختمان داده مناسب برای الگوریتم های جستجو، طراحی کنید.

تمرین ۲-۲: انواع مسائل زیر را همراه با مثال، توضیح دهید.

- 1- Single-state problems, 2-Multiple-state problems, 3-Contingency problems, 4-Exploration problems