

## ۴- جست‌وجوی عمقی محدود شده (Depth-Limited Search)

در الگوریتم جست‌وجوی عمقی محدود شده (DLS) گسترش گره‌ها حداکثر تا عمق  $L$  (که به عمق برش معروف است) انجام خواهد شد. در این روش با در نظر گرفتن یک محدوده عمقی مانند  $L$ ، مشکل مسیر بی‌نهایت حل می‌شود. حال اگر کم‌عمق‌ترین هدف بعد از عمق برش قرار گیرد (یعنی  $L < d$  باشد)، این روش جست‌وجو، کامل نیست و اگر  $L \geq d$  باشد، جست‌وجوی عمقی محدود شده کامل است، اما همچنان بهینه نیست. گاهی اوقات عمق برش براساس دانش مسأله تعیین می‌شود. برای مثال در نقشه رومانی چون ۲۰ شهر وجود دارد، بنابراین طول راه حل حداکثر ۱۹ خواهد بود. لذا با انتخاب  $L = 19$  هیچ وقت گره‌ای با عمق بیش از ۱۹ بررسی نخواهد شد. خصوصیات این الگوریتم به شرح زیر است:

۱- کامل نیست؛ زیرا اگر  $L < d$  باشد جوابی پیدا نمی‌کند.

۲- بهینه نیست؛ زیرا اگر  $L > d$  باشد نیز هیچ تضمینی برای یافتن راه حل بهینه وجود ندارد.

۳- پیچیدگی زمانی این روش برابر است با  $O(b^L)$ .

۴- پیچیدگی حافظه این روش برابر است با  $O(bL)$ .

۵- برای استفاده از این استراتژی باید بتوان عمق جواب را به درستی حدس زد، اما در برخی از مسائل، تعیین عمق مناسب دشوار است.

نکته ۷: جست‌وجوی اول عمق نوع خاصی از جست‌وجوی عمقی محدود شده با  $L = \infty$  است.

شبهه کد جست‌وجوی عمق محدود شده بازگشتی در شکل زیر آمده است.

**Function** DEPTH\_LIMITED\_SEARCH (problem, limit) **return** a solution-or failure cutoff  
**return** RECURSIVE\_DLS (MAKE\_NODE (INITIAL-STATE [problem]), problem, limit)

**Function** RECURSIVE-DLS (node-problem, limit) **return** a Solution, or failure cutoff  
 cutoff-occurred?  $\leftarrow$  false  
 If GOAL-TEST[problem](STATE [node]) then **return** Solution (node)  
 else if DEPTH [node] = limit then **return** cutoff  
 else for each Successor in EXPAND (node, problem)  
     result  $\leftarrow$  RECURSIVE-DLS (successor, problem, limit)  
     if result = cutoff-occurred?  $\leftarrow$  true  
     else if result  $\neq$  failure then **return** result  
 if cutoff-occurred? Then **return** cutoff else **return** failure

پیاده‌سازی بازگشتی جست‌وجوی عمق محدود

## ۵- جست‌وجوی عمقی تکرار شونده (Iterative Deepening Search)

جست‌وجو با عمق محدود همواره با مشکل تعیین عمق برش مناسب مواجه است؛ زیرا ممکن است راه حل مسأله در عمقی بزرگ‌تر از عمق برش قرار داشته باشد. جست‌وجوی عمقی تکرار شونده (IDS) در واقع جست‌وجوی عمقی محدود شده است که برای یافتن بهترین عمق برش چندین بار اجرا می‌شود، جست‌وجوی عمقی تکرار شونده این کار را با افزایش آهسته عمق برش انجام می‌دهد، ابتدا ۰، ۱، ۲، و... تا یک هدف پیدا شود. الگوریتم این روش در شکل زیر نشان داده شده است.

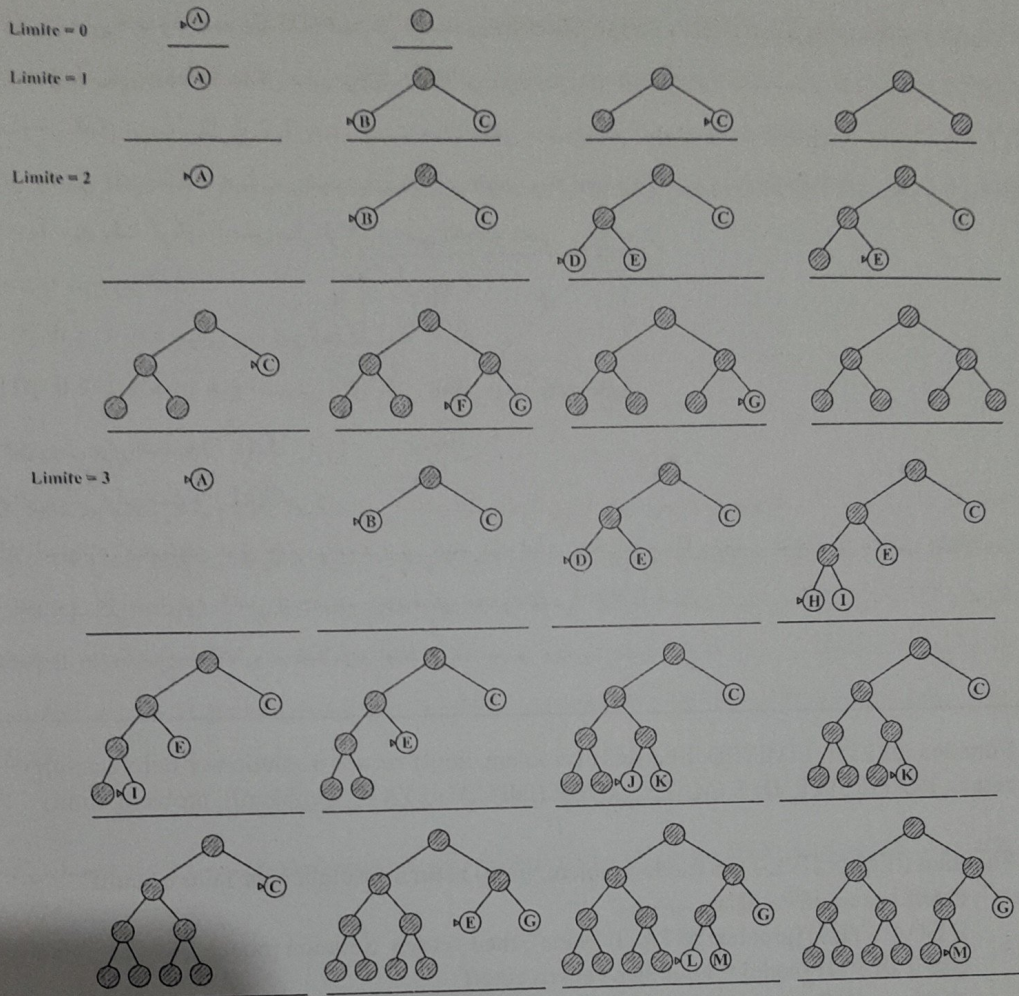
**Function** ITERATIVE-DEEPENING-SEARCH (problem) **return** a Solution, or failure  
 inputs: problem, a problem  
 for depth  $\leftarrow$  0 to  $\infty$  do  
     DEPTH-LIMITED-SEARCH (problem, depth)  $\leftarrow$  result  
     if result  $\neq$  cutoff then **return** result

الگوریتم جست‌وجوی عمقی تکرار شونده





شکل زیر مراحل اجرای جست و جوی عمقی تکرارشونده را برای عمق برش  $L=0$ ,  $L=1$ ,  $L=2$  و  $L=3$  نشان می دهد.



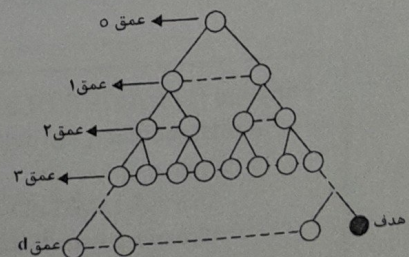
چهار تکرار برای جست و جوی عمقی تکرارشونده

خصوصیات این الگوریتم به شرح زیر است:

- ۱- کامل است؛ زیرا تمام سطوح درخت را به صورت کامل بررسی می کند (البته به شرطی که فاکتور انشعاب متناهی باشد).
- ۲- بهینه است؛ زیرا هر سطح را قبل از سطوح بعدی آن بررسی می کند (البته به شرطی که وزن همه یال ها برابر باشد).
- ۳- تعداد کل گره های تولید شده برابر است با  $N(IDS) = d(b) + (d-1)b^2 + \dots + (1)b^d$ . در جست و جوی عمقی تکرارشونده، گره ها در آخرین سطح (عمق  $d$ ) ۱ بار، در سطح مقابل آن (عمق  $d-1$ ) ۲ بار و... تولید می شوند؛ بنابراین پیچیدگی زمانی این روش  $O(b^d)$  است؛ زیرا در درخت جست و جویی که در همه سطوح ها فاکتور انشعاب یکسان است، بیشتر گره ها در آخرین سطح قرار دارند؛ بنابراین خیلی مهم نیست که گره های سطوح بالاتر چندین بار تولید می شوند. البته در شرایطی این حالت برقرار است که فاکتور انشعاب برابر ۱ نباشد.

عمق	حداکثر تعداد گره	دفعات تکرار
۰	۱	$d+1$
۱	$b$	$d$
۲	$b^2$	$d-1$
۳	$b^3$	$d-2$
$\vdots$	$\vdots$	$\vdots$
$d$	$b^d$	۱

حداکثر تعداد گره ها در هر سطح از درخت و تعداد دفعات تولید هر گره در جست و جوی IDS





با توجه به شکل فوق تعداد کل گره‌های تولید شده برابر است با:

$$N(IDS) = (d+1)b^0 + (d)b^1 + (d-1)b^2 + \dots + b^d = O(b^d)$$

حال اگر این تعداد گره‌ها را با گره‌های تولید شده در روش BFS مقایسه کنیم داریم:

$$N(BFS) = b + b^2 + \dots + b^d + (b^{d+1} - b) = O(b^{d+1})$$

بنابراین می‌توان گفت جست‌وجوی BFS، بعضی گره‌های عمق  $d+1$  را تولید می‌کند، درحالی‌که جست‌وجوی IDS آن‌ها را تولید نمی‌کند؛ در نتیجه جست‌وجوی عمقی تکرار شونده، علی‌رغم تولید حالات تکراری، سریع‌تر از جست‌وجوی اول سطح می‌باشد.  
به‌عنوان مثال اگر  $b=10$  و  $d=5$  باشد داریم:

$$N(IDS) = 50 + 4000 + 3000 + 200000 + 1000000 = 123450$$

$$N(BFS) = 10 + 100 + 1000 + 10000 + 100000 + 1000000 + 999990 = 1111100$$

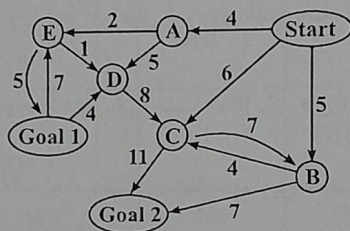
۴- پیچیدگی حافظه این روش مانند جست‌وجوی اول عمق بوده و برابر  $O(bd)$  می‌باشد.

۵- این جست‌وجو مزایای جست‌وجوی عمقی و سطحی را با هم ترکیب می‌کند.

۶- مرتبه گسترش حالات، مشابه جست‌وجوی سطحی است، با این تفاوت که بعضی از حالات چندین بار بسط داده می‌شوند.

۷- اگر فضای حالت بزرگ و عمق راه‌حل نامشخص باشد، در میان جست‌وجوی ناآگاهانه، جست‌وجوی عمقی تکرار شونده ترجیح داده می‌شود.

مثال ۷: برای درک بیشتر الگوریتم، (IDS) مثال ۳ را به روش (IDS) برای رسیدن به اولین Goal پیمایش می‌کنیم.



پاسخ: ☒

محدودیت عمق = 0

Start

محدودیت عمق = 2

Start, A, D, E, B, C, G<sub>2</sub>

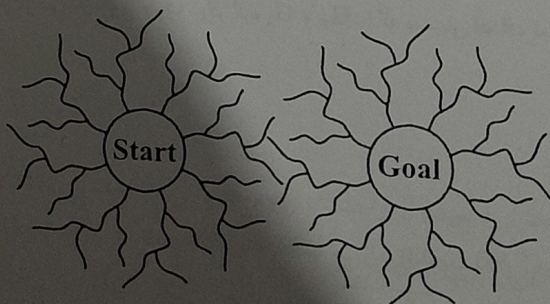
محدودیت عمق = 1

Start, A, B, C

توجه: عیب این روش آن است که تعداد زیادی گره به‌صورت تکراری بررسی می‌شوند، پس در این قسمت سربار (Overhead) زیادی خواهیم داشت.

## ۶- جست‌وجوی دوطرفه (Bidirectional Search)

در الگوریتم‌های جست‌وجوی قبلی، جست‌وجو همواره از یک گره به‌عنوان ریشه درخت یا مبدأ شروع می‌شد و تا رسیدن به هدف جلو می‌رفت، ولی در بسیاری از مسائل، هدف از قبل شناخته شده است. در این گونه موارد، جست‌وجو می‌تواند هم از مبدأ به سمت مقصد و هم از مقصد به سمت مبدأ پیش برود تا در این میان به گره مشترکی برسد. در واقع این روش، نوع تغییر یافته‌ای از جست‌وجوی اول سطح است که در دو سمت از مبدأ به هدف (یعنی روبه جلو یا forward) و در جهت عکس از هدف به سمت مبدأ (یعنی رو به عقب یا backward) جست‌وجو انجام خواهد شد و زمانی الگوریتم متوقف می‌شود که هر دو سمت در میانه راه به هم برسند. به عبارت دیگر، زبان پایان الگوریتم، یافتن گره‌ای است که هم در پیمایش روبه جلو و هم در پیمایش رو به عقب پیمایش شود. شکل زیر شماتیک جست‌وجوی دوطرفه را نشان می‌دهد.



شماتیک جست‌وجوی دوطرفه