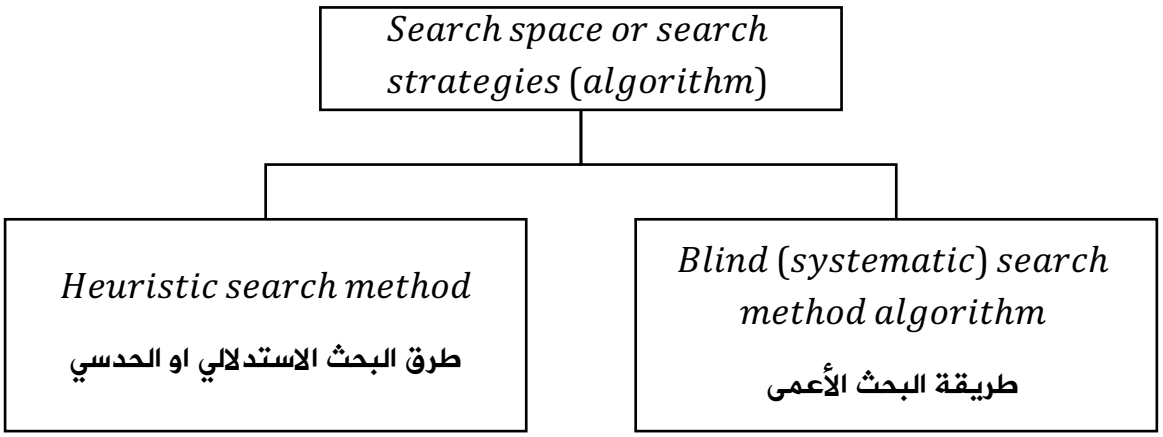
بسم الله الرحمن الرحيم

استراتيجيات البحث او خوارزميات البحث (طُرق البحث)

Search strategies (Algorithms)

يمكن تعريف البحث على انه اختبار منظم او مهيكل لقضايا الحالة للعبور على المسار من حالة البداية إلى حالة المستهدفة. تم استخدام عدة طُرق لتمثيل عملية البحث. في ذلك الرسم البيان والشجرة والرسم وما إلى ذلك. وتقسم طُرق البحث إلى ثلاثة أجزاء رئيسية:

1) *Blind (systematic) search method algorithm.* **(طريقة البحث الأعمى)**
2) *Heuristic search method.* **(طرق البحث الاستدلالي او الحدسي)**
3) *optimization methods.* **(طرق البحث الذي من خلالها نصل الى الهدف)**



هو البحث الذي يبحث في جميع الاحتمالات الممكنة (يجب ان تكون هدف) وتقسم إلى:	ممكن ان توجد الهدف او لا توجد ويعتمد فقط على حالة الكلفة وتقسم إلى
1) طريقة (خوارزمية) البحث بعمق (البحث العمودي) *Depth−first search Algorithm*	1) التسلق الشاهق *Hill climbing*
2) طريقة البحث المستعرض والعرضي (البحث العمودي) *Breadth−first search Algorithm*	2) خوارزمية البحث الافقي أولاً أو اللحظي *Best−first search Algorithm*
3) هجين (عمق + اتساع) بحث *Hybrid (Depth +Breadth) Search*	3)خوارزمية A^* *AlgorithmA^**

لكل طريقة من الطرق:

1) شرح الطريقة 2) خوارزمية 3) مثال عددي 4) مثال تطبيقي 5) خصائص 6) فروقات

أسئلة

لماذا سمي بالبحث الاعمى؟ لأنه يبحث في العقد ..... لماذا سمي بالبحث الحدسي؟ لأنه لا يمر بجميع العقد.

ملاحظة/ المستويات تبدأ من الصفر دائماً

الفرق بين (طريقة البحث الأعمى) و(طريقة البحث الحدسي (الاستدلالي))		
ت	طريقة البحث الأعمى	طريقة البحث الحدسي (الاستدلالي)
1	هذه الطريقة أبطأ او اقل سرعة من طريقة البحث أو الاستدلال للوصول الى الهدف	هذه الطريقة أسرع من طريقة البحث الاعمى للوصول الى الهجف
2	مضمونة للوصول الى الحل	غير مضمونة الوصول الى الحل
3	لا تعتمد على مقادير الكُلف (عقد ومسارات)	تعتمد على مقادير الكُلف (عقد ومسارات)
4	تقسم الى $Depth$ و $Breadth$	تقسم الى $Hill\ Climbing$ و $B.F.S$ و A^*
5	تستغرق وقت وجهد وخزن أكثر من $Heuristic$ عند الوصول الى الهدف	تستغرق وقت وجهد اقل من
6	نبحث في الـ $Depth$ عمقاً يسارًا بشكل عمودي وفق الـ $Breadth$ مستوى مستوى وبشكل افقي	نبحث عن الأقل كُلفة
7	البحث الأعمى يحتاج الى جهد وتستغرق وقت وخزن ومكلفة جداً..... الـخ عند الوصول الى الهدف	لا تستغرق وقت وجهد
8	نهر دجلة	خارطة كندا

الفرق بين الـ $Graph$ و الـ $Tree$

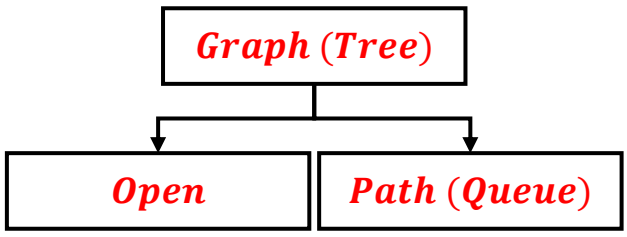
في $Tree$ لا يوجد $Sycle\ path$ أي مسار دائري.

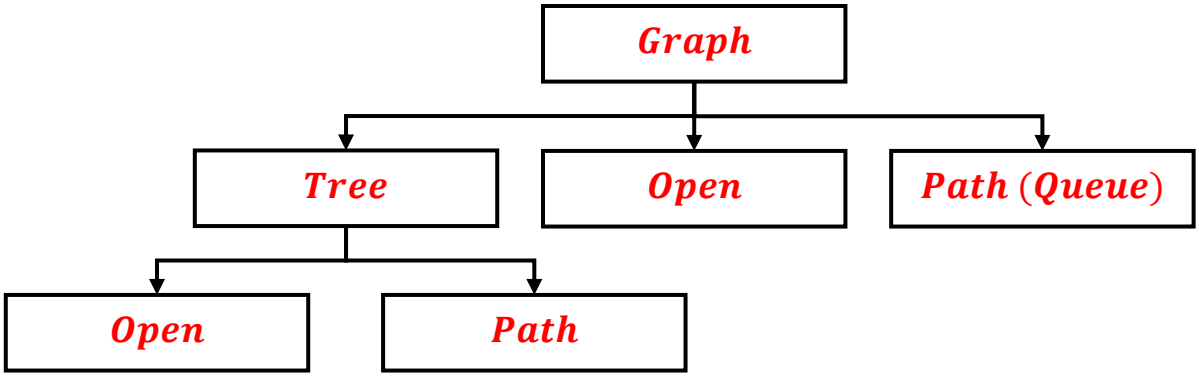
عندما نحول الـ $Graph$ الى $Tree$ نبدأ بـ A وننتهي بـ A.

أي $Graph$ أو $Tree$ يمكن ان يصل بطريقتين:

1- طريقة $Open$ الجدول (يتم استخدام الجدول).

2- طريقة $Path$ (الطوابير).





اي **Graph** يمكن تحويله الى **Tree** ويصل بطريقة الـ **Open** و طريقة الـ **Path** وكذلك الـ **Tree**

ت	*Open*	*Path*
1	طريقة تستخدم الجدول	طريقة لا تستخدم الجدول
2	تعتمد على رسم الشجرة والـ *Graph*	تعتمد على الرسم فقط وتأخذ المسار الأقل
3	تحتوي على *Open* و *Close*	لا تحتوي على *Open* و *Close*
4	يكون حلها النهائي قمة و فرع من الشجرة (الشجرة)	يكون حلها النهائي على شكل فرع من الشجرة
5	لا تلغي المسارات	تلغي كل المسارات أو (كل الطرق الغير مسلوكة) او الغير مستخدمة
6	تبحث عن الطريق الأقل كُلفة في A^*	تبحث عن الأقل كلفة في جميع الطرق

يتم تقييم الاستراتيجيات (الطُرق) وفقاً للأبحاث التالية (ايهما أفضل):

1- الاكتمال $(completes)$: هل هناك حل موجود (هل دائماً تجد حل).

2- تعقيد الوقت $(time\ complexity)$: هو عدد العقد المتولدة (أو التي تتم إنشاءها).

3- تعقيد الفضاء $(Space\ complexity)$: الحد الأقصى لعدد العقد في الذاكرة أي اكبر عدد من العقد المتراكمة.

4- الامثلية $(optimality)$: الامثلية في إيجاد دائماً الحل الأقل كلفة (هل تجد دائماً الحل الأقل كُلفة)

طُرق (خوارزميات) البحث الاعمى النظامية او البحث التام $Blind\ Search\ methods\ (Algorithms)$

فضاء البحث هو استكشاف دون الاستفادة من أي معلومات على المسألة (المشكلة) ايضاً تسمى البحث غير المنتظم.

الطُرق البارزة:

1- طريقة خوارزمية البحث بعمق (البحث العامودي) $Depth-First\ Search\ Method\ (Algorithm)$

2- طريقة خوارزمية البحث المستعرض الموسع (البحث الافقي) $Breadth-First\ Search\ Method\ (Algorithm)$

تعتمد طريقة الاختيار لطرق (خوارزمية) البحث الاعمى على توقعنا لموقع الهدف، فاذا كان الهدف قريب من نقطة البداية فإننا الاختيار الأفضل هو طريقة (خوارزمية) البحث المستعرض (البحث الافقي)، واذا كان توقع الهدف بعيد عن نقطة البداية فان الاختيار الأفضل هو طريقة (خوارزمية) البحث بعمق (البحث العمودي) أي ان الفرق بينهم (اذا كان الهدف قريب عن $(Start)$ أو بعيد عن $(Start)$)

1- طريقة خوارزمية البحث بعمق (البحث العامودي) $Depth-First\ Search\ Method\ (Algorithm)$

تعتبر من اهم طرق خوارزميات البحث الاعمى فكرتها هي البحث باتجاه العمق بمسار واحد من جهة اليسار الى ان نصل الى نقطة الهدف او نصل لنقطة ليس لها ابن، في هذه الحالة نبدأ عملية العودة $Back\ traprocks$ الى الأعلى وذلك لــ اختيار (فحص) الأنماط الأخرى. تستخدم في الحل مجموعتين هما $Open$ و $Close$ كلاهما يتم إضافة وحذف العناصر منهما من جهة اليسار.

$Open$: عمود (خانة) تحتوي على العقد التي سنبحث عنها للحل الى ان نصل للهدف

$Closed$: خانة تحتوي على كل العقد التي بحثنا فيها عن الهدف ولم نجده (لم نجد الهدف)

ادخال العقدة الأولى هو اخراج العقدة الأخيرة.

هذه الخوارزمية توصل للحل بسرعة اذا كان الهدف بعمق حيث انها كفؤة البحث في تفرعات كثيرة.

خوارزمية البحث بعمق تعتمد على البحث العمودي (بعمق) للوصول الى الهدف وتستخدم اذا كان الهدف بعمق حتى تختصر الوقت. تعتمد على مجموعتين (مصفوفتين) $Open$ و $Closed$ مبدأ عملهم يشبه $Stack$ (تمثيل البيانات في الذاكرة).

مبدأ (نوع) $Stack$ الذي تستخدمه خوارزمية البحث بعمق يسمى $First\ In\ First\ Out\ (FIFO)$ يعني (اول واحدة تدخل اخر واحدة اخرج) وعملية البحث في الشجرة تبدأ من جهة اليسار

ملاحظة: إن من أهم طرائق البحث الاعمى هو البحث بعمق.

في هذه الخوارزمية يتم استخدام مجموعتين هما $Open$ و $Closed$ كلاهما فيه حذف واضافة من جهة اليسار

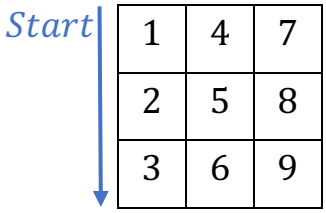
ويمكن اعتبار مجموعتي $Open$ و $Closed$ من ضمن صنف التكديس وهذ الخوارزمية تستخدم مبدأ

$(Last\ In\ First\ Out)$ **LIFO** , $(First\ In\ First\ Out)$ **FIFO**

ان فكرة هذه الخوارزمية هي الاتجاه نحو مسار واحد حتى تضل اكثر عمقاً ونبدأ من اليسار الى اليمين.

1- آلية عمل ال $Depth$ هي البحث عمقاً الى اليسار دائماً (تبحث عمقاً الى اليسار) وتضاف الأبناء من جهة اليسار.

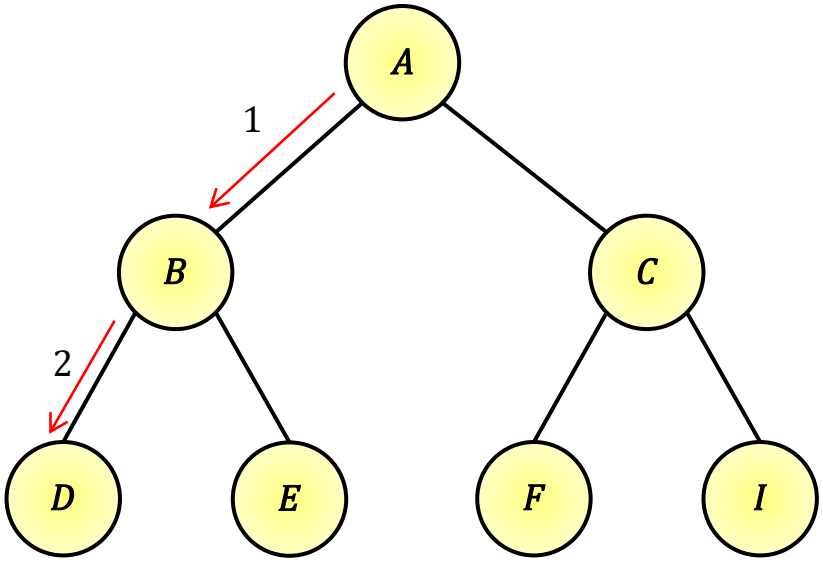
2- كيف تبدأ من اليسار. كما في الشكل الاتي:



3- ماهي المتطلبات (الأسئلة المتواردة في هذه الطريقة)

a) يعطي مخطط بياني (شجرة، رسم بياني)

b) نبدأ من $Start$ ونتجه الى اليسار نزولاً:



c) المتطلبات هي:

c_1) $State\ Space$: فضاء الحالة للحل وهو كل النقاط او جميع العقد في الشجرة التي دخلت في عمود (مصفوفة) الـ $Open$

c_2) $Search\ Space$: فضاء البحث أي جميع الحالات (العقد) التي تم فحصها في عمود او مصفوفة الـ $Open$.

اخر شيء في عمود $Close$ بمعنى العقد التي المرور عليها للبحث عن الحل (أي كل النقاط التي تم زيارتها واختبارها او فحصها هل تمثل مجال او لا؟

c_3) $Solution\ Space$: مسار الحل هو اقصر طريق (المسار) للحل من نقطة الجذر الى نقطة الهدف يوصل الى الحل.

d) عمل جدول نجد به ما يلي:

$Iteration$	$Open$	$Close$
0	$Start$	$\emptyset$
1	⋮	⋮
n	⋮	n

نبدأ بعمل جدول يحتوي عمود التكرار (الحالات) ويحتوي على:

d_1) عمود $Open$: يحتوي على أعمدة البداية وبقية العقد التي ستبدأ البحث فيها عن الهدف

d_2) عمود $Close$: يحتوي على العقد التي تم البحث فيها عن الهدف ولم نجدها

e) في كل عقدة نسأل:

هل هي الـ $Goal$ ← نتوقف	هل لها أبناء؟ نستمر	هل هي $Closed$

مميزات أو خصائص البحث الأول بعمق $Depth-First\ Search\ properties$

1) تستخدم عندما يكون طريق الهدف بعيد عن نقطة البداية (نقطة البداية بعيدة عن الهدف).

2) العقدة التي تدخل اولاً تخرج (العقدة الاولى تدخل تخرج اول واحدة أي ادخال العقدة الأولى هو اخراج العقدة الأخيرة).

3) في الطريقة يكون البحث عمقاً يساراً (الى اليسار) يعني (AB).

4) ادخال عقد جديدة (الأبناء) من جهة اليسار أي العقدة الجديدة تدخل من جهة اليسار (الأبناء).

5) يحتوي على أسلوب البحث بعمق (البحث العمودي) على عملية التراجع الخلفي.

6) سريعة الوصول الى الحل.

7) كفؤة للبحث في تفرعات كثيرة وذلك لأنها لا تحتفظ بكل العقد في المسار المعطى في عمود ال $Open$

8) الأمثلة هي الألعاب مثل الشطرنج والدومينا والورق.

مساوئ الخوارزمية: غير مضمونة لإيجاد أقصر طريق للحل

تعليمات الخوارزمية $Depth - First\ Search\ Algorithm\ (FIFO)$ مهمة جداً

1) $Put\ the\ start\ node\ in\ open$
2) $Loop: if\ open\ is\ empty\ then\ exit\ (fail)$
3) $N: first\ (open)$
4) $If\ goal\ (n)\ then\ exits\ (success)$
5) $Generate\ a\ children\ of\ n$
6) $Remove\ (n, open), add\ (n, closed)$
7) $Expand\ n\ by\ replacing\ that\ node\ by\ left\ of\ open$
8) $Go\ to\ loop\ in\ step\ 2$

1- نضع عقدة البداية في $Open$

2- $Loop$: اذا كان $open$ فارغ اذاً فشل تم الخروج

3- N: هو أولا في $open$

4- اذا كانت (n) هي الهدف اذاً نجح البحث (تحقق النجاح) وخروج

5- هل الـ n لها أبناء (تولد أبناء من n) اضفها في يسار الـ $close$

6- نبدأ بإزالة n $(n, open)$ وتضاف n الى $(n, closed)$

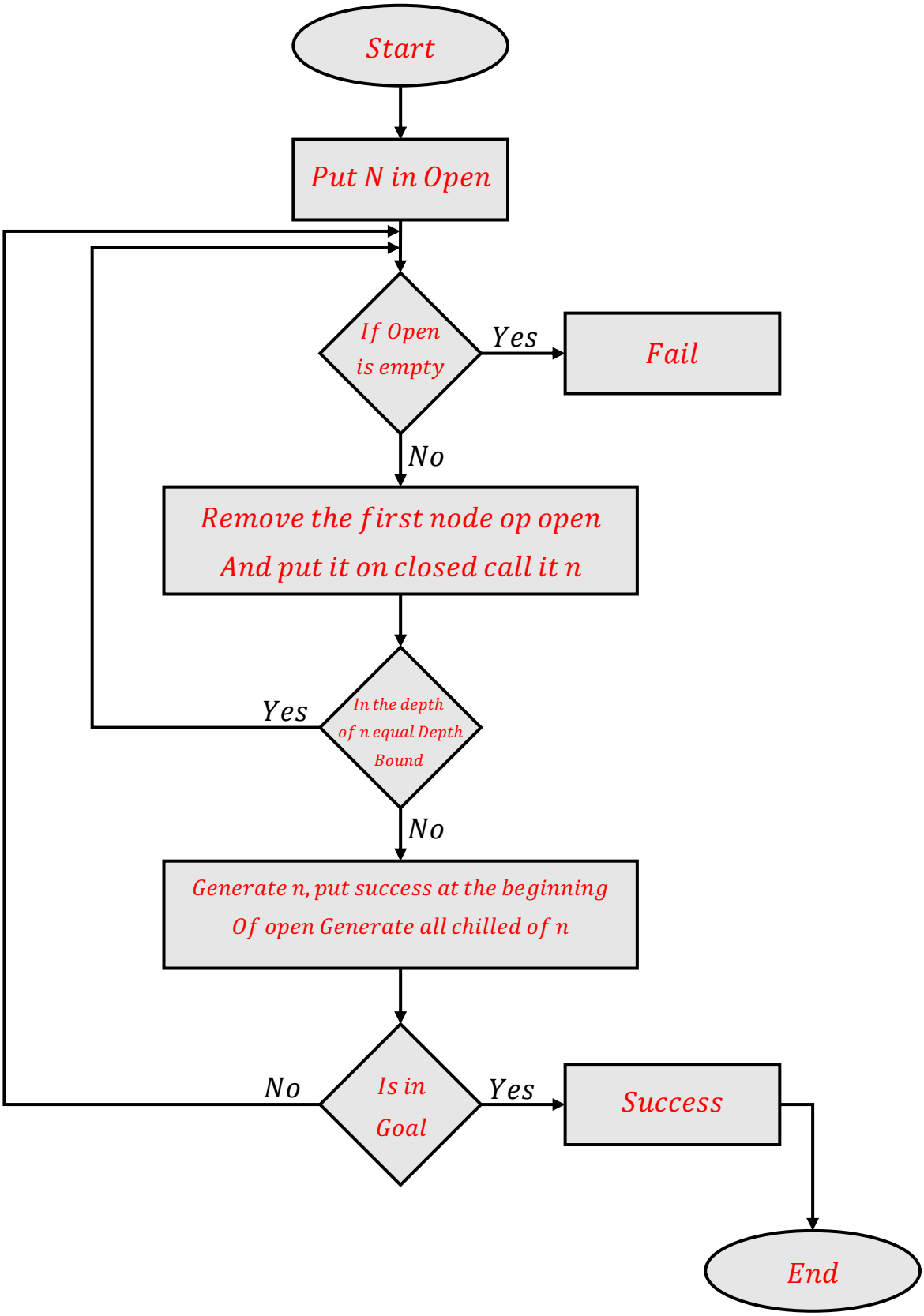
7- توسيع n وذلك عن طريق استبدال العقد (الأبناء) من اليسار (بواسطة اليسار المفتوح)في $open$

8- اذهب الى $Loop$ (الخطوة الثانية).

المخطط الانسيابي لخوارزمية البحث بعمق (مهم جداً)

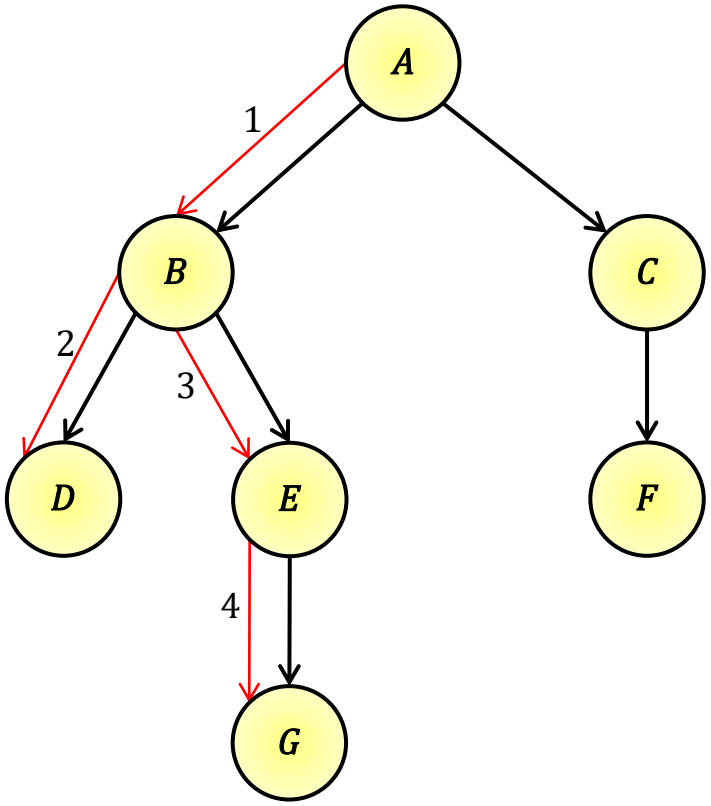
$Flowchart\ of\ depth - first\ search\ algorithm$

في الصفحة القادمة


Start
Put N in Open
If Open is empty
Yes
Fail
No
Remove the first node op open And put it on closed call it n
In the depth of n equal Depth Bound
Yes
No
Generate n, put success at the beginning Of open Generate all chilled of n
Is in Goal
No
Yes
Success
End

مثال:

Start state: ***A***, *Goal state*: ***G***



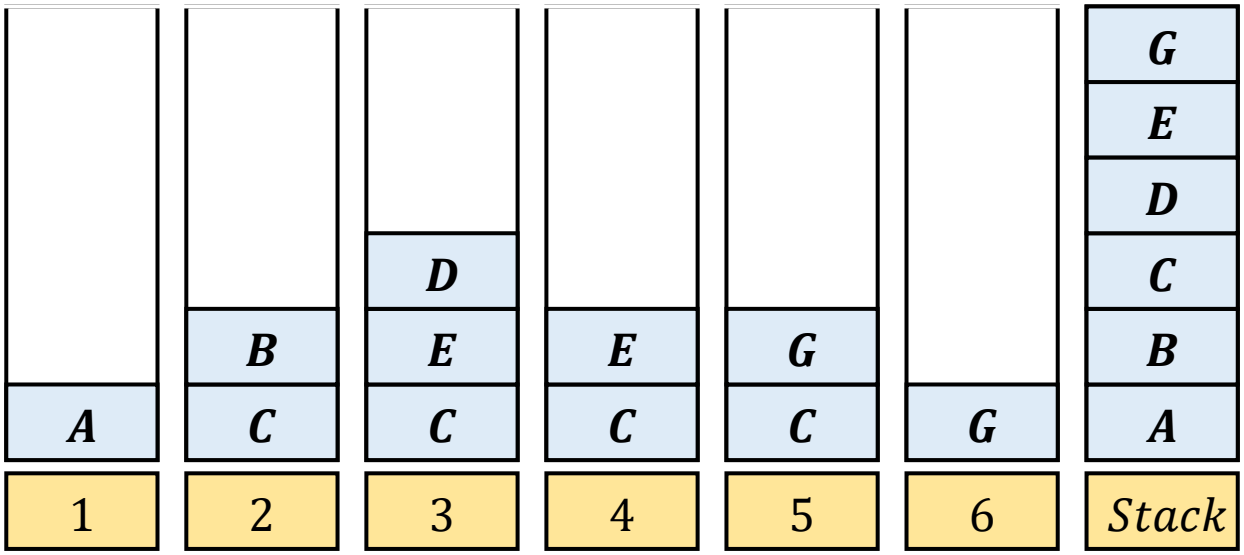
Solution:

شرط أن نبدأ عمقاً من اليسار

Iteration	*Open*	*Closed*
1	$[A, 0]$ *or* A	[ ] *or* $\emptyset$
2	$[(B, A), (C, A)]$	$[(A, 0)]$*or* A
3	$[(D, B), (E, B), (C, A)]$	$[(A, 0), (B, A)]$
4	$[(E, B), (C, A)]$	$[(A, 0), (B, A), (D, B)]$
5	$[(G, E), (C, A)]$	$[(A, 0), (B, A), (D, B), (E, B)]$
6	***G*** *is the Goal*	$[(A, 0), (B, A), (D, B), (E, B), (G, E)]$

كيفية تمثيل البيانات في ذاكرة الحاسوب $Stack$ باستخدام مبدأ $(FIFO)$ أو $(LIFO)$

مراحل تحديث $Updating$ للـ $Stack$ أثناء حل خوارزمية البحث بعمق: ↓



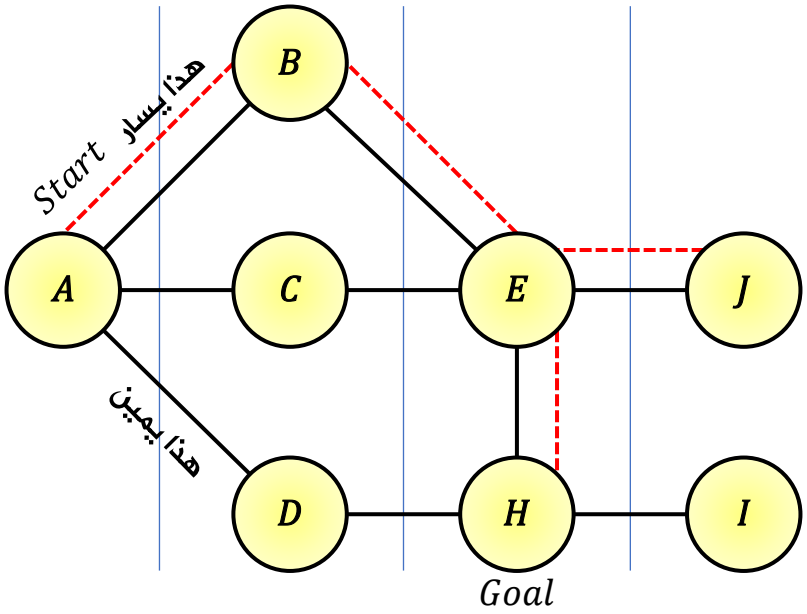
$$Start\ space: A$$

$$Search\ space: A, B, D, E, G$$

$$Solution\ path: A\ B\ E\ G \quad \textbf{or} \quad A \rightarrow B \rightarrow E \rightarrow G$$

بِسْمِ اللَّهِ الرَّحْمَٰنِ الرَّحِيمِ

استخدم خوارزمية البحث بعمق $(Depth - First\ Search\ Method\ (Algorithm))$ اذا علمتَ أن عقدة البداية هي (A) وعقدة النهاية هي (H):



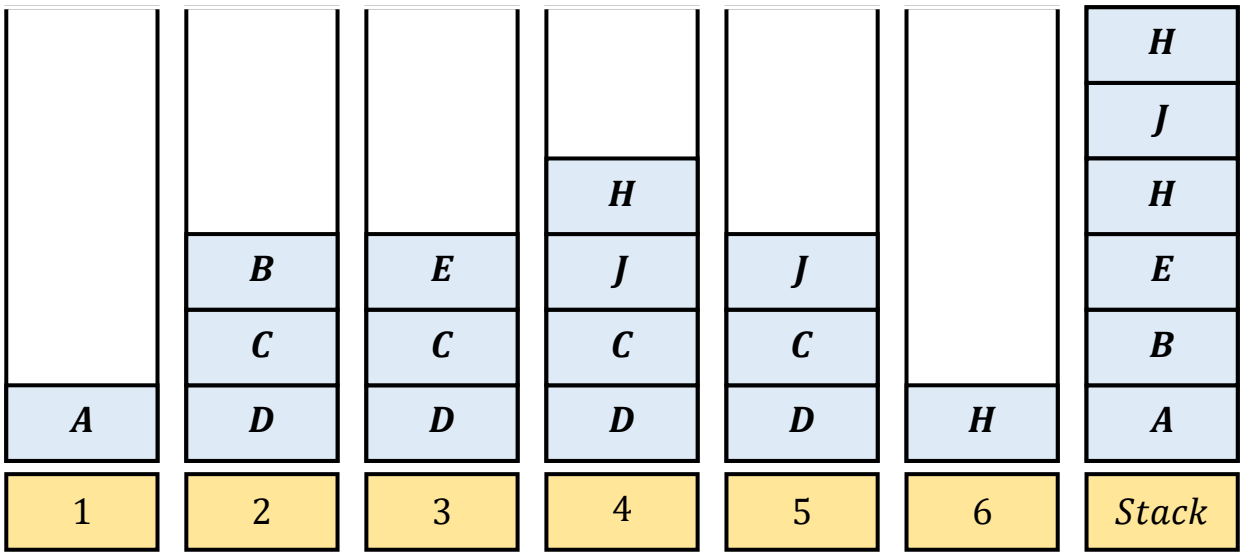
$Start\ node$: A
$Goal\ node$: G

ملاحظة الترتيب بـ $(Open)$ والـ $(Closed)$ مهم جداً.

Iteration	Open	Closed
1	A	$[\ ]\ or\ \emptyset$
2	$B\ C\ D\ or\ [(B,A),(C,A),(D,A)]$	$[(A,0)]\ or\ A$
3	$E\ C\ D\ or\ [(E,B),(C,A),(D,A)]$	$[(A,0),(B,A)] or\ A,B$
4	$H\ J\ C\ D$	$A\ B\ E$
5	$J\ C\ D$	$A\ B\ E\ H$
6	$H\ is\ The\ Goal$	

ملاحظة: في التكرار (4) لا نقوم بالتحويل (لأنه لم نمر عليه بالهدف)

The Stack:

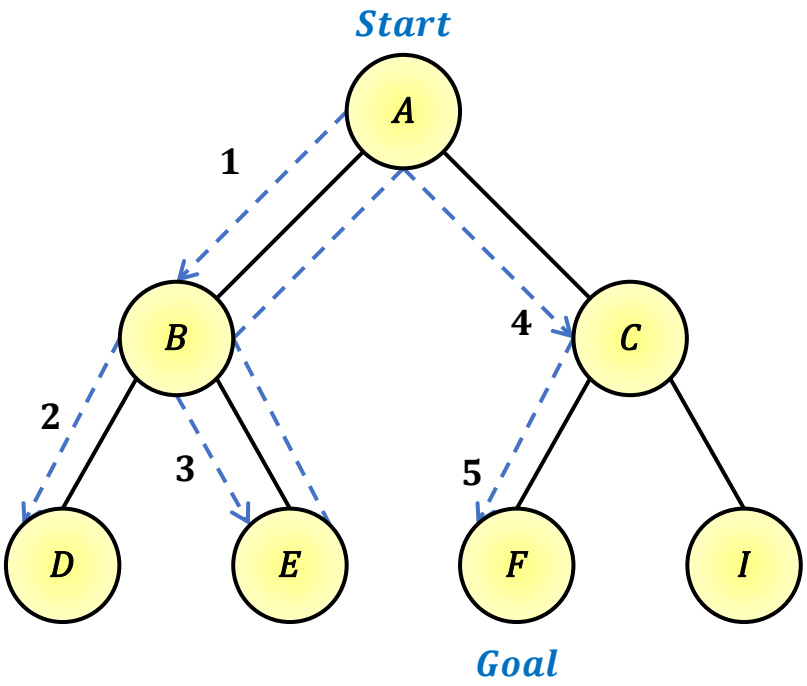


Start Space: ***A***

Search Space: ***A, B, C, D, E, H***

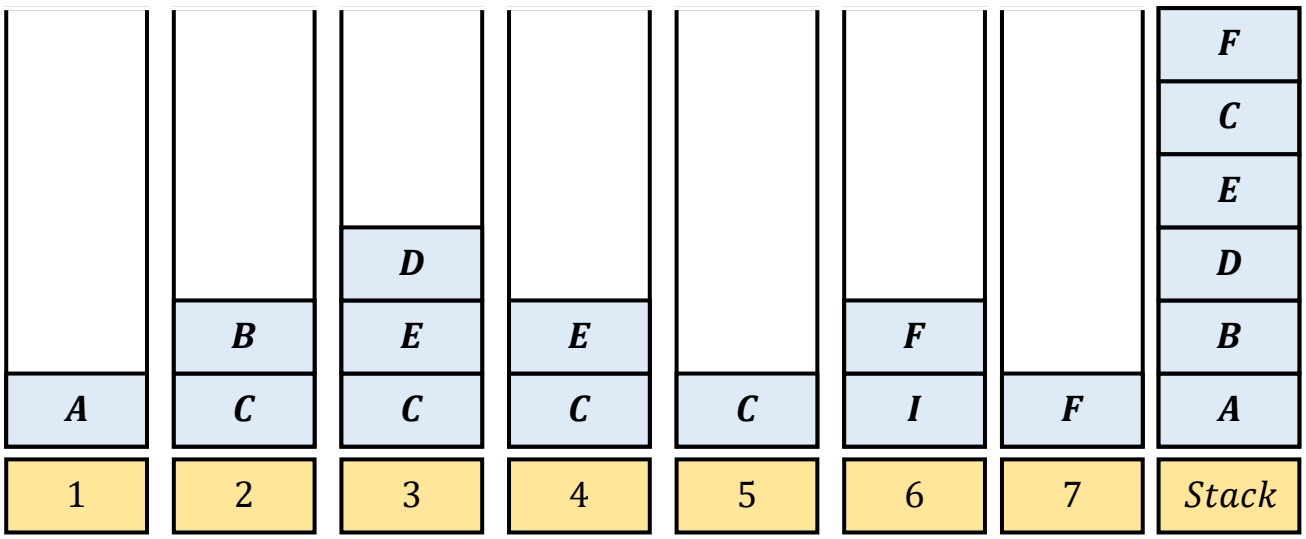
Solution Path: ***A − B − E − H***

مثال: استخدم خوارزمية البحث بعمق *(Depth − First Search Method (Algorithm))* **اذا علمت أن عقدة البداية هي** (A) **وعقدة النهاية هي** (F)**:**



Iteration	Open	Closed
1	A	$[\ ]\ or\ \emptyset$
2	$B\ C$	A
3	$D\ E\ C$	$A\ B$
4	$E\ C$	$A\ B\ D$
5	C	$A\ B\ D\ E$
6	$F\ I$	$A\ B\ D\ E\ C$
7	***F*** *is The Goal*	

The Stack:



Start Space: ***A***

Search Space: ***A, B, C, D, E, F***

Solution Path: ***A − C − F***

تنفيذ خوارزمية البحث بعمق على مسألة الـ *8 − Puzzle*

1) *Start node*
2) *Goal node*
3) *Search method*

ملاحظة: العقد التي فيها تكلفة أعلى تـــهمل.

ملاحظة: في هذه الطريقة يعطي المستوى التي نتوقف فيه بالحل عند تطبيق خوارزمية البحث بعمق على هذه المسألة.

سوف يطلب في السؤال التوقف عند المستوى ونبدأ بالرجوع الى بقية المستويات الى ان نصل الى الهدف.

ملاحظة: يجب كتابة تسلسل العقد مع المستويات ويساراً.

مثال: نفذ خوارزمية البحث بعمق على مسألة الـ $Puzzle - 8$ اذا علمتَ أن الـ $Start\ node$ والتوقف علن المستوى 3

الحل: