

Search Algorithm
(Informed Search)

د. ياسر خضرا

محتوى مجاني غير مخصص للبيع التجاري

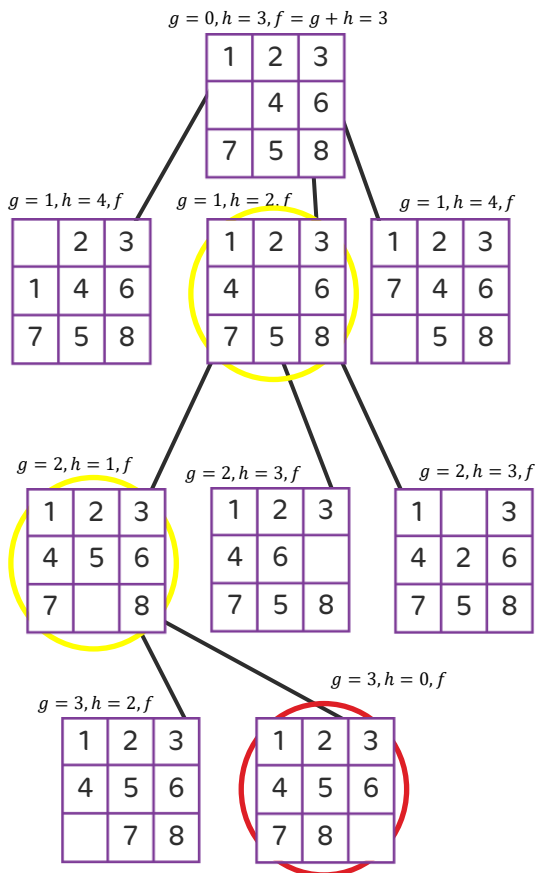
09/05/2023

RB Informatics; مبادئ الذكاء الصناعي

في المحاضرة السابقة بدأنا بالحديث عن طرق البحث التي يمكن للوكيل الذكي استخدامها لحل المشاكل والمهام المطلوبة منه ، وتعرفنا على أول نوع لخوارزميات البحث وهي الخوارزميات العمياء Uninformed/Blind Search ، وسنكمل في محاضرتنا مع النوع الثاني منها وهو الخوارزميات الموجهة Informed Search .

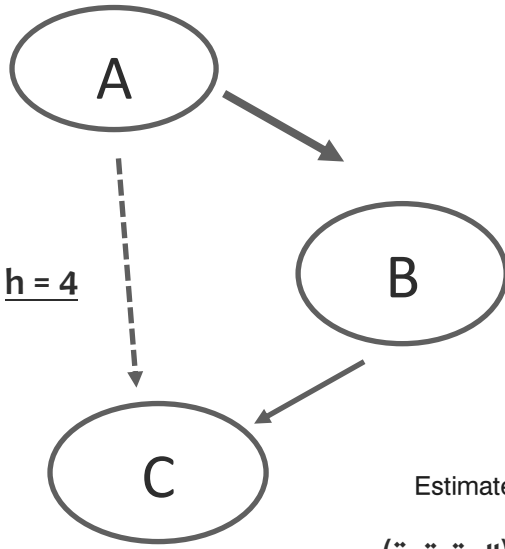
■ الخوارزمية الذكية: هي خوارزمية مستخدمة في الذكاء الصناعي أو هي خوارزمية يستخدمها الوكيل الذكي لاتخاذ القرارات المناسبة حسب الهدف المطلوب .

Informed Search Algorithm



- خوارزميات البحث الموجهة تكون مفيدة عندما يكون لدينا فضاء بحث كبير Large Search Space مثل : مجمع مشافي أو سلسلة شركات أو غيرها .
- تستخدم هذه الخوارزميات فكرة الاستكشاف والاسترشاد لذلك يطلق عليها اسم البحث الاسترشادي Heuristic Search .
- لو عدنا لمثال The 8-Puzzle في البحث الأعمى كنا ندرس جميع الاحتمالات الممكنة حتى نصل للحالة المثلى.
- بينما في البحث الاسترشادي أو الموجه سنقوم بعملية مقارنة بين الحالة الحالية (حالة البداية) والحالة المطلوبة (الهدف) من خلال تابع مقارنة سيقوم بإحصاء المربعات المتواجدة بمكانها الصحيح والمتبقية بشكل عشوائي ونصل للهدف المطلوب عندما يصبح عدد المربعات العشوائية صفر.
- من الممكن أن نجد الهدف بأكثر من فرع للشجرة وبالتالي سنبحث عن الطريق الأقصر والأقل تكلفة للوصول له.

تابع الاسترشاد Heuristic Function



- الاسترشاد أو الاستدلال هو تابع يُستخدم في الخوارزميات الموجهة ويعمل على إيجاد المسار الصحيح بأقل تكلفة للوصول للهدف المطلوب .
- يعتمد الحالة الراهنة للوكيل كدخل له ثم يقوم بتقدير مدى قربها من الهدف .
- الطريقة الاستكشافية من الممكن ألا تساعدنا بالوصول لأفضل حل Best Solution لكنها تضمن لنا أن نحصل على حل جيد بوقت معقول وفق العلاقة : $h(n) \leq h^*(n)$

$h(n)$ تكلفة الاسترشاد , Heuristic Cost , $h^*(n)$ تكلفة مقدرة Estimated Cost

- لذلك يجب أن تكون كلفة الاسترشاد أصغر أو تساوي التكلفة المقدرة (المتوقعة).

البحث الاسترشادي الصافي Pure Heuristic Search

- البحث الاسترشادي الصافي هو أبسط شكل لخوارزميات البحث الاسترشادي (ليس لديه أي تحسين على خوارزميته).
- يزيد عدد العقد بناءً على قيمتها الاسترشادية $h(n)$.
- يملك قائمتين Open List و Closed List .
- في الـ Closed List نضع العقد التي تم استكشافها بالفعل ووجدنا أنها تحقق الهدف أو تؤدي لتحقيقه وفي الـ Open List نضع العقد التي لم يتم استكشافها بعد.
- في كل تكرار لعملية البحث نختار العقدة n صاحبة أقل قيمة استرشاد (استدلال) $h(n)$.
- نقوم باستكشافها وإدخال كل أبنائها (العقد المرتبطة معها لـ Open List ثم ننقل العقدة n لـ Closed List ، وتستمر الخوارزمية حتى يتم العثور على الهدف المطلوب .



"Turn your wounds into wisdom."

-Oprah Winfrey

Best-First Search Algorithm (Greedy Search)



- خوارزمية الـ Greedy Best-First Search دائماً تختار المسار الذي يبدو أفضل في حالة ما.
- هي عبارة عن خليط من خوارزميتي الـ Bfs و Dfs حيث تتيح لنا الاستفادة من مزايا كلا الخوارزميتين السابقتين..
- تستخدم تابع الاسترشاد والبحث $h(n)$.
- تساعدنا هذه الخوارزمية على اختيار العقدة المناسبة بكل خطوة.
- في هذه الخوارزمية نكتشف العقدة الأقرب للهدف (أو لعقدة الهدف) والأقرب
- تكلفة لما هو مقدر في تابع الاسترشاد Heuristic Function وفق العلاقة التالية : $f(n) = h(n)$
- حيث $h(n)$ هي التكلفة المقدرة من العقدة n للهدف.
- يتم تنفيذ خوارزمية The Greedy Best-First Algorithm بالاعتماد على The Priority queue.

خطوات تنفيذ خوارزميات الـ Best-First Search

1. نضع عقدة البداية Start في الـ Open List .
2. إذا كانت الـ Open List فارغة تتوقف الخوارزمية وتعيد نتيجة أنه تم الفشل .
3. ننقل العقدة صاحبة أقل تكلفة استرشاد $h(n)$ من الـ Open List لـ Closed List .
4. نستكشف العقدة الأقل تكلفة ونضيف أبنائها لـ Open List .
5. نقوم باختبار العقد الجديدة (الأبناء) لنعلم إن كانت تحقق الهدف أم لا . إذا كانت أي منها هي الهدف عندها نعيد نتيجة أنه تم النجاح وتتوقف عملية البحث ، وإلا ننتقل للخطوة 6 .
6. من أجل كل عقدة سيتم إضافتها لاحقاً ، الخوارزمية ستتحقق من تابع الـ $F(n)$ وستقوم بالبحث إذا ما كانت هذه العقدة موجودة سابقاً في Open or Closed List . إذا لم تكن العقدة موجودة في كلا القائمتين عندها نضيفها لـ Open List .
7. نعود للخطوة 2 .



"Run when you can, walk if you have to, crawl if you must; just never give up."

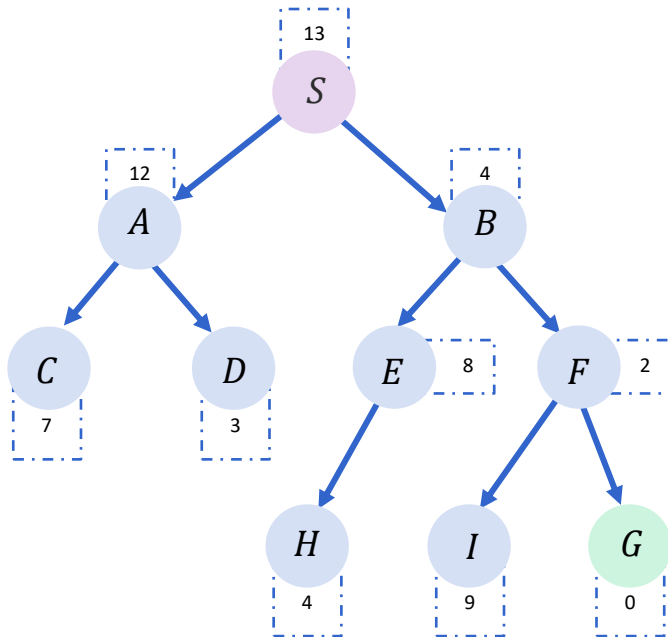
-Dean Karnazes

- لو أخذنا مثال لعبة السودوكو ، وكما نعلم فهي تتكون من تسع مربعات بداخل كل مربع تسع مربعات (أي 81 مربع صغير) تحوي الأرقام من (1 إلى 9) ، ولها قواعد معينة لنصل لنهايتها .
- لو تم تمثيل اللعبة بشجرة بحث $= <$ سيكون عدد الأرقام الموجودة بمكانها الصحيح هو قيمة عقدة البداية وقيم باقي العقد يتحدد أيضاً وفق عدد مربعات الأرقام الموجودة بمكانها الصحيح .
- والهدف المطلوب هو الوصول بأقصر طريق وأقل عدد خطوات لتكون ال 81 مربع معبئة بطريقة صحيحة .
- فلو كان ببداية اللعبة 28 مربع موجود بمكانها الصحيح ، لن نعبر عنها ب 28 عقدة بل مرحلة البداية كاملة سنسميها العقدة A وتكون قيمتها 28 ويكون المتبقي هو (عدد العناصر الصحيحة - 81) ومن ثم نناقش جميع الحالات الممكنة التي تساعدني بالوصول للهدف .
- فكل ما زاد عدد الأرقام الموجودة بمكانها الصحيح تتغير قيمة العقد المتفرعة عن العقدة A ، فمن الممكن أن تتفرع A لعقدتين إحداهما تزيد عنها بمقدار 2 والثانية بمقدار 3 ، فنختار التي تساعدنا بالوصول للهدف النهائي أسرع ، وهكذا تستمر الخوارزمية حتى نجد أقصر طريق للحل .
- وتمثل الحالات الممكنة بالرموز A,B,C,..... أي أن العقدة الواحدة في شجرة البحث تعبر عن حالة كاملة تمت دراستها ، كما مر معنا بمثال The 8-Puzzle أيضاً .
- ونضع عند كل عقدة من هذه العقد تابع التكلفة الخاص بها $h(n)$.

مثال :

لنلاحظ شجرة البحث الموضحة أمامنا ونستخدم خوارزمية greedy Best-First Search في التجوال والوصول لعقدة الهدف فيها .

- عقدة الانطلاق (البداية) Start نرمز لها S .
- عقدة الوصول للهدف Goal نرمز لها G .

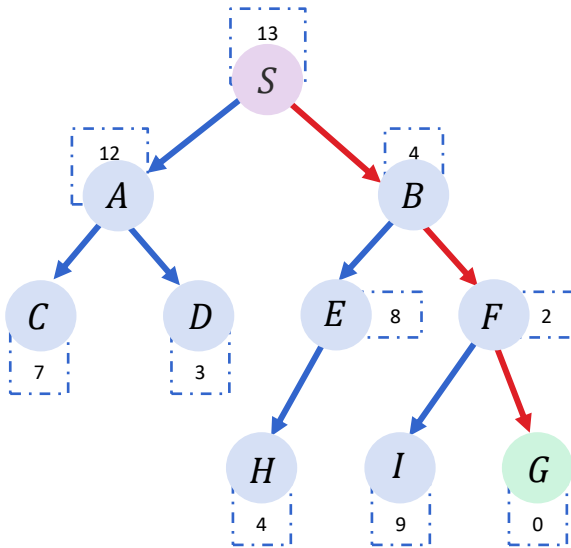


تذكير بأنواع الأقواس :

- { قوس مجموعة للقوائم List والتابع .
- [] قوس مصفوفة .
- () للعمليات الحسابية .
- ⌈ تقريب لأعلى قيمة .
- ⌊ تقريب لأدنى قيمة .

node	H(n)	node	H(n)
A	12	B	4
C	7	D	3
E	8	F	2
H	4	I	9
S	13	G	0

في كل تكرار للخوارزمية سوف نختبر العقدة حسب تقدير تابع التكلفة وفق العلاقة $f(n) = h(n)$ كما موضح بالجدول .

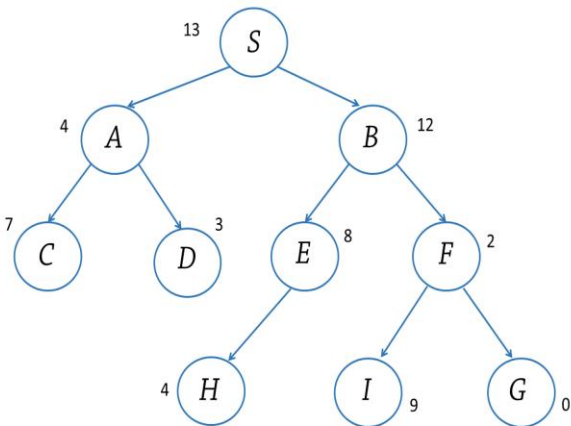


step	exploring	open	close
initialization		{S}	{}
1	S	{A, B}	{S}
2	B	{E, F, A}	{S, B}
3	F	{G, I, E, A}	{S, B, F}
4	G	{I, E, A}	{S, B, F, G}

$S \rightarrow B \rightarrow F \rightarrow G$

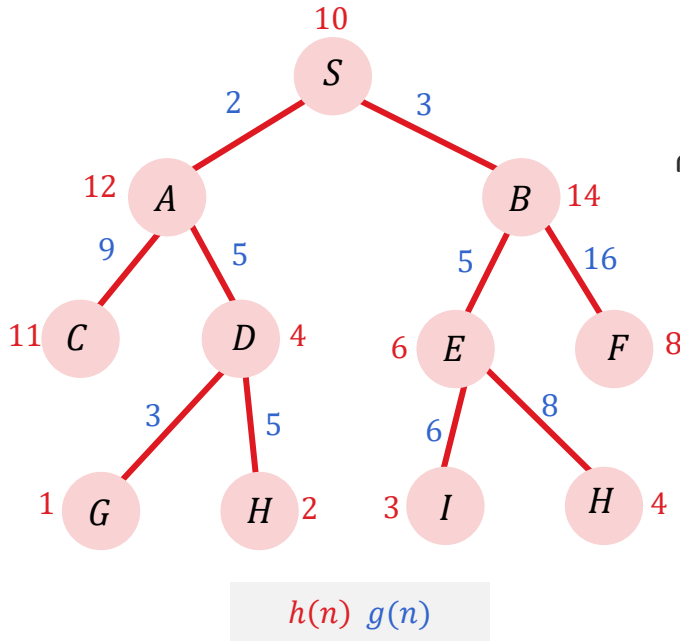
- نفتح قوس مجموعة عند ال Open ونضع فيه ال S .
- Close ما زالت مغلقة ، والمسار الذي سألبحث عنه سيكون موجود بال Close .
- Exploring تحوي العقد التي يتم اختبارها الآن (تحت المجهر) .
- في البداية لم يكن بإمكاننا إلا أن نشاهد العقدة S ولكن عندما دخلت لقائمة ال Open تمكنا من رؤية أبنائها ومن شجرة البحث نجد أنهما العقدتين A,B وبالتالي سنضيفهم ل Open list وننقل العقدة S ل Closed list .
- نعود للعقدتين A,B وبالنظر لشجرة البحث نجد أن تكلفة ال A هي $h(A) = 12$ ال B تكلفتها $h(B) = 4$ ، وكما اتفقنا سابقاً سنختار الكلفة الأقل وبالتالي ال B ستدخل الاختبار Exploring وباستكشافها نجد أن أبنائها العقدتين E,F ونضيفهم ل Open list إلى جانب العقدة A وننقل العقدة B ل Closed list ويكون المسار الذي وصلنا له حتى الآن هو {S,B} .
- والآن لدينا ال {E,F,A} بالنظر لتابع الكلفة لكل منهم نجد كلفة E : $h(E) = 8$ ، F : $h(F) = 2$ وحسب الخوارزمية سننتقل ل F وتدخل ال Exploring ونستكشف أبنائها فنجد العقدتين G,I ونضيفهم ل Open list لجانب ال E,A وننقل العقدة F ل Closed List .
- والآن نعود لاختبار العقد في ال Open list : ونجد أن العقدة G تحقق الهدف فتتوقف الخوارزمية .
- وننقل ال G ل Closed list ويكون المسار الذي حصلنا عليه هو $S \Rightarrow B \Rightarrow F \Rightarrow G$.
- ونلاحظ أن المسار وفق الخوارزميات الموجهة تحدده التكلفة الأدنى بينما في الخوارزميات العمياء نبحث بكل المسارات حتى نصل للهدف .

حالة افتراضية للمثال السابق :



لو كانت قيم A,B معكوسة وكانت الشجرة بالشكل التالي :
 نلاحظ أننا سننتقل ل A بالمرحلة الثانية ويصبح المسار {S,A}
 وبعدها سنختار D وهنا سنصل لنهاية الشجرة وفق هذا المسار
 {S,A,D} ولكننا لم نصل للحل أو الهدف المطلوب لذلك تحدثنا
 سابقاً أن هذا النوع من الخوارزميات لا يضمن وجود حل دائماً.

A* Search Algorithm



خوارزمية ال A* هي النوع الأكثر شيوعاً من خوارزميات

Best-First Search .

تعتمد على تابع الاسترشاد $h(n)$ ، وتكلفة الوصول للعقدة n

من الحالة الابتدائية $g(n)$.

تجد خوارزمية ال A* أقصر مسار للوصول للهدف

باستخدام تابع ال $h(n)$.

خوارزمية البحث هذه تستكشف قسم قليل من الشجرة،

وتساعدنا بالحصول على الحل الأمثل بشكل أسرع.

في خوارزمية البحث A* : نستخدم البحث الاسترشادي

بالإضافة لتكلفة الوصول للعقدة .

ثم نجمع كلا الكلفتين معاً ويدعى هذا المجموع : fitness number

$$F(n) = g(n) + h(n)$$

Estimated Cost Of The
Cheapest solution

Cost To Reach Node(n)
From Start State .

Cost To Reach From Node(n)
To Goal Node.

وكما نشاهد في شجرة البحث الموضحة أعلاه العقدة S تكلفتها 10 والمسار الذي ستسير وفقه لتصل ل B كلفته 3 والعقدة B كلفتها 14 والمسار الذي ستسير وفقه لتصل ل F كلفته 16 وهكذا..

تعريفات:

ليكن لدينا شجرة البحث التالية وتابع كلفة كل عقدة بها :

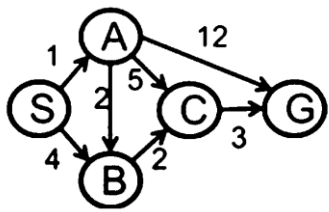
نفترض المسار التالي : $S \rightarrow A \rightarrow C$.

لنجد كلفة الوصول عبر هذا المسار نحتاج ما يلي وفق خوارزمية ال A* :

g وهي كلفة الانتقال من S إلى A إلى C (أي حساب المسارات وجمعهم) ، $g(1+5=6)$.

H وهي قيمة تابع الاسترشاد h لآخر عقدة في المسار وهي C في

مثالنا ونجدها في الجدول السابق .



Node	H
S	7
A	6
B	4
C	2
G	0

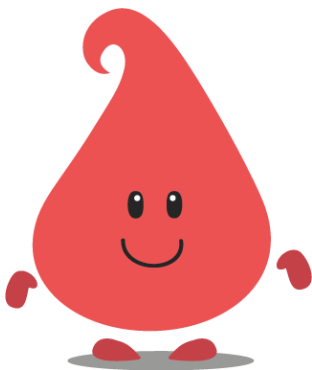
■ نجمع القيمتين لسابقتين لنحصل على التكلفة المقدرة لهذا المسار :

$$F=g + h=6+2=8$$

- يمكن تشبيه طريقة عمل هذه الخوارزمية بطريق سفرنا من دمشق إلى اللاذقية مثلاً **إما** أن أسير داخل مدينة حمص ثم طرطوس ثم أصل إلى اللاذقية وأكون بهذه الطريقة استهلكنا زمن ووقود ومسافة بلا هدف مطلوب أو فائدة (مسار طويل) .
- **أو** يمكننا ان اعتمد الطريق المخصص للسفر وأمر من جانب كلا المدينتين عبره (ويمثل هذا الطريق المسار في شجرة البحث) ثم أسير داخل مدينة اللاذقية حتى أصل للمكان الذي أريده (وهذا يمثل تكلفة عقدة النهاية).

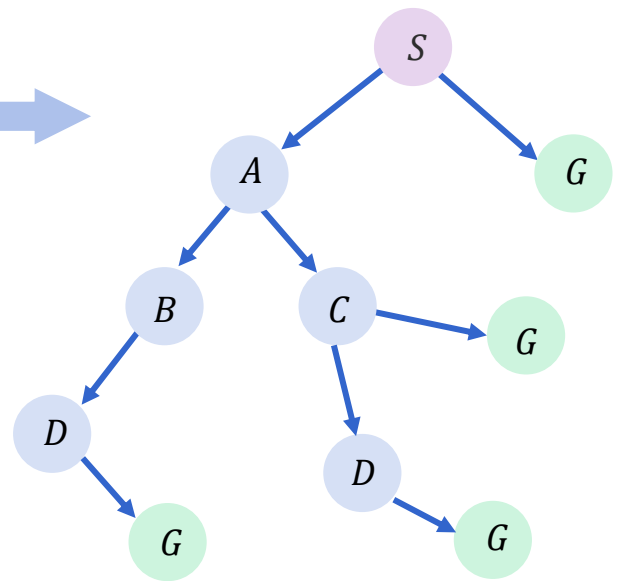
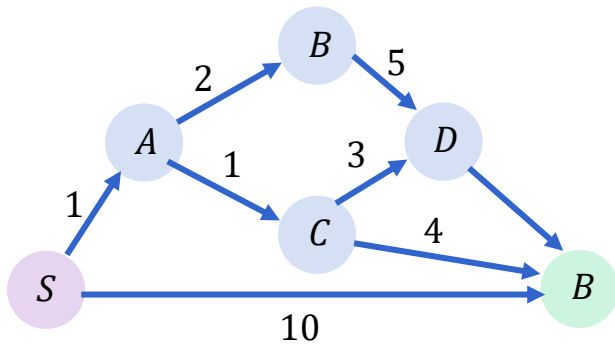
ما مر معنا حتى الآن يمكننا من استنتاج خطوات سير خوارزمية ال A* :

1. نضع عقدة البداية في ال Open List.
2. إذا كانت ال Open List فارغة نتوقف الخوارزمية وتعيد نتيجة أنه تم الفشل.
3. نختار من ال Open List العقدة n صاحبة أقل قيمة لتابع التقييم ($f=g + h$) إذا كانت العقدة n هي الهدف المطلوب عندها نعيد نتيجة أنه تم النجاح ونتوقف الخوارزمية وإلا ننتقل للخطوة 4.
4. نستكشف العقدة n وننظر لكل أبنائها وننقل العقدة n ل Closed List من أجل كل عقدة من أبنائها n نختبر إذا كانت موجودة مسبقاً في ال Open Or Closed List، إذا لم تكن موجودة سنقوم بحساب تابع التقييم لكل ابن من أبنائها n ونضيفه ل Open List.
5. إذا كان أحد أبنائها n موجود مسبقاً في Open Or Closed List عندها يجب أن تكون مرفقة بمؤشر خلفي back pointer يشير لأقل قيمة لتابع كلفة مسار هذه العقدة الابن $g(n)$.
6. نعود للخطوة 2.



"Prepare for the worst, hope for the best."

■ مثال: في الشبكة أو شجرة البحث الموجودة أمامنا



state	$h(n)$	state	$h(n)$
S	5	C	2
A	3	D	6
B	4	G	0

عقدة البداية لها S والهدف أو عقدة النهاية هي G ونلاحظ أنه يمكننا الوصول لها بأكثر من طريق إما من خلال المسار $S \rightarrow A \rightarrow C \rightarrow D \rightarrow G$ أو $S \rightarrow G$ أو $S \rightarrow A \rightarrow C \rightarrow G$ ومسارات أخرى أيضاً، لذلك وللسهولة عندما نجد شبكة مغلقة نحاول أن نقوم بفردتها حتى تتوضح لنا أكثر، ونلاحظ أن تابع كلفة الاسترشاد لكل عقدة موضح بالجدول أعلاه، وكلفة المسار لكل عقدة موضحة على الشجرة.

والمطلوب هنا أن نجد أفضل مسار وفق خوارزمية A* للوصول من S إلى G.

أول ما سيخطر لنا أن نختار الطريق المباشر ($S \rightarrow G$) لكن لو عدنا لنلاحظ كلفة المسار فيه وكلفة العقدة G سنجد

$$F = g + h, \quad f = 10 + 0 = 10$$

نبدأ بالتنفيذ وفق خطوات خوارزمية A* :

- في مرحلة التهيئة initialization سنضع : $\{(S, S)\}$
- نبدأ بتكرار تنفيذ الخوارزمية Iteration :

1. Iteration 1 : $\{(S \rightarrow A, 4), (S \rightarrow G, 10)\}$

■ من S لدينا مسارين من ($S \rightarrow G$) و ($S \rightarrow A$) ، ننظر لتكلفة كل مسار :

$$F = g + h$$

$$S \rightarrow A : \quad F_a = 1 + 3 = 4$$

$$S \rightarrow G : \quad F_g = 10 + 0 = 10$$

■ بالمقارنة نجد أن الأفضل لنا أن تذهب للعقدة A .

2. Iteration 2 : $\{(S \rightarrow A \rightarrow C, 4), (S \rightarrow A \rightarrow B, 7), (S \rightarrow G, 10)\}$

■ حيث من العقدة A لدينا مسارين ل B,C

$$S \rightarrow A \rightarrow B = (1+2)+4=7$$

$$S \rightarrow A \rightarrow C = (1+1) + 2=4$$

■ ونجد أن الطريق الأفضل هو $(S \rightarrow A \rightarrow C)$

■ وبالنظر للعقدة C نجد أنها تتفرع لمسارين إما مع G أو مع D .

3. Iteration 3 : $\{(S \rightarrow A \rightarrow C \rightarrow G, 6), (S \rightarrow A \rightarrow C \rightarrow D, 11), (S \rightarrow A \rightarrow B, 7), (S \rightarrow G, 10)\}$

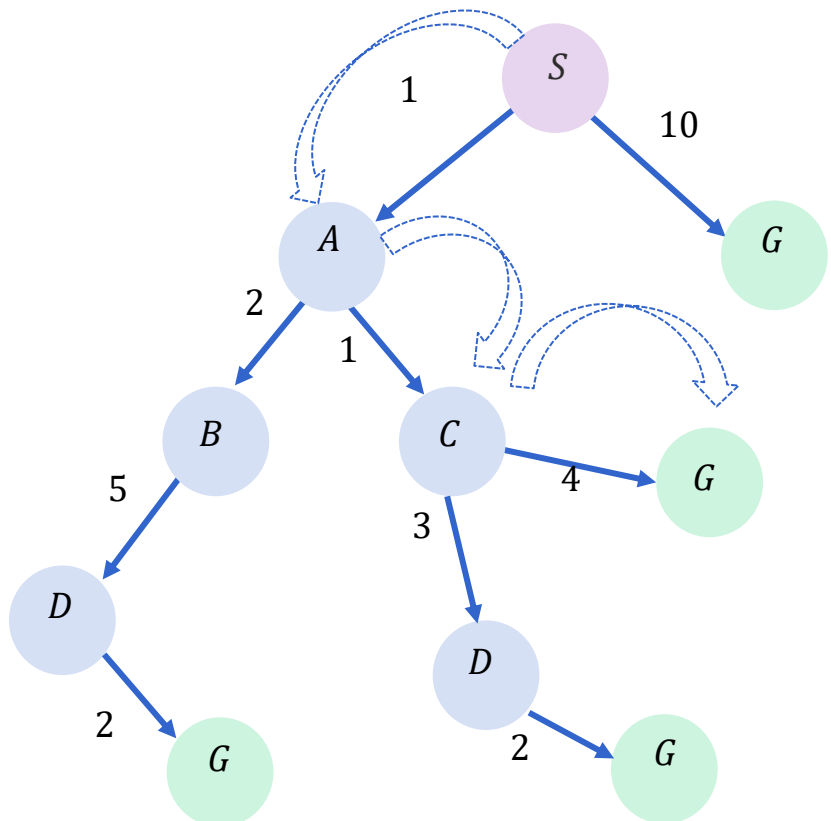
$$S \rightarrow A \rightarrow C \rightarrow G = (1+1+4) + 0 = 6$$

$$S \rightarrow A \rightarrow C \rightarrow D = (1+1+3) + 6 = 11$$

■ ونلاحظ أننا وصلنا للهدف ونجحت مهمة الخوارزمية A*

4. Iteration 4 : will give the final result , as $S \rightarrow A \rightarrow C \rightarrow G$

■ أي عند التكرار الرابع للخوارزمية سنعيد النتيجة الأخيرة وهي المسار الذي يحقق الهدف المطلوب .



: The 8-Puzzle Example ↩

حتى نطبق خوارزمية A* هنا سنحتاج g و h :

h : هي عدد المربعات الموجودة بمكان خاطئ والذي نحدده بالمقارنة بين الحالة الراهنة والهدف المطلوب.

أو من خلال جمع مسافة منهاتن distance

Manhattan بين العقد التي في غير مكانها. (مسافة منهاتن هي المسافة بين نقطتين تشكلان ضلعا المثلث القائم).

g : هو عدد العقد التي تجاوزناها من عقدة البداية حتى وصلنا للعقدة الحالية.

أو تعريف آخر: هو رقم المستوى بشجرة البحث فعند

عقدة البداية المستوى صفر وقيمة g تساوي الصفر.

وعندما نحرك الفراغ إما لليمين أو اليسار أو الأعلى أو

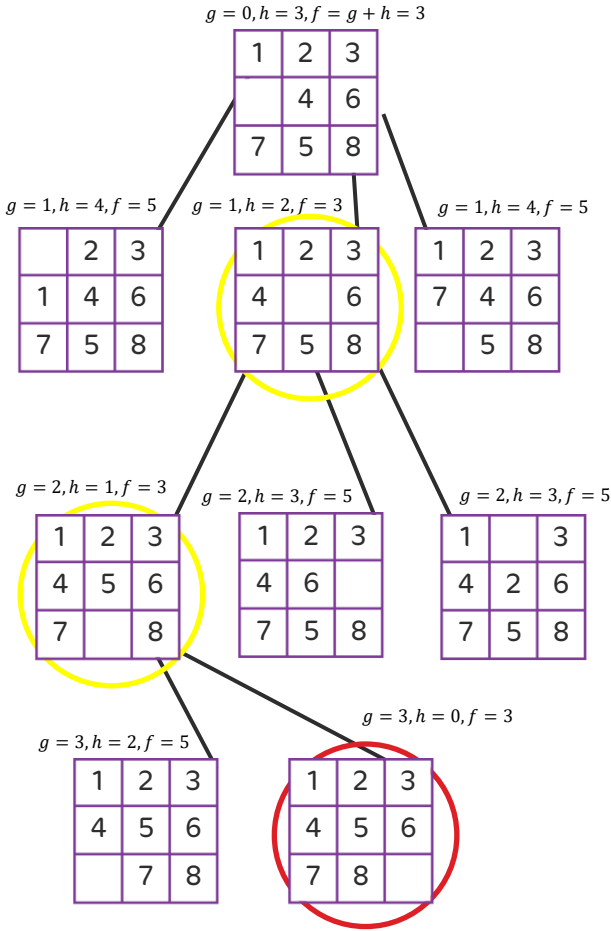
الأسفل نكون قد أصبحنا بالمستوى الأول وقيمة g

تساوي الواحد وبعدها ننتقل للمستوى الثاني حيث $g=2$

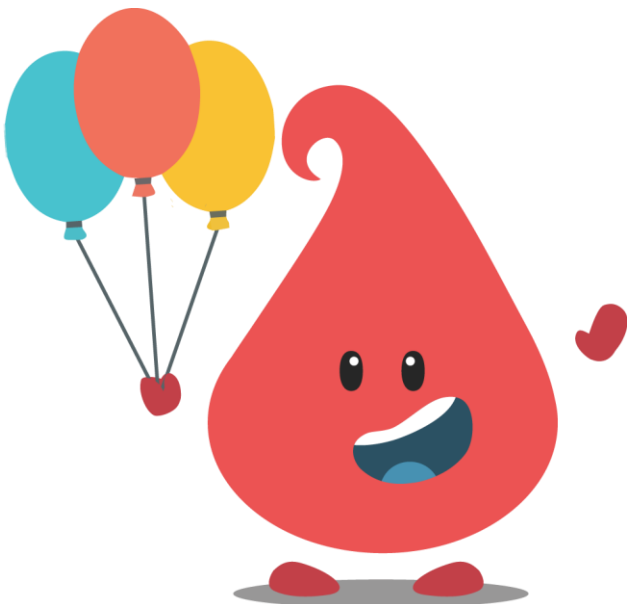
ثم للمستوى الثالث $g=3$. بعد الانتهاء مما سبق

نستبدل الشبكة السابقة بالرموز (S,A,B,.....,C) لتعبر

عن كل حالة استكشفناها بطريقة أسرع وأسهل.



وفي النهاية نود التنويه أن هذه الخوارزميات تستخدم في تطبيقات ال GPS لتحديد المسارات وحل ألعاب الذكاء مثل 8-Puzzle و السودوكو وغيرها



The End