كلية الهندسة
قسم الهندسة المعلوماتية

مقرر خوارزميات بحث ذكية

د. غزوان علي ريّا

محاضرات الأسبوع الثاني
الفصل الأول 2024–2025

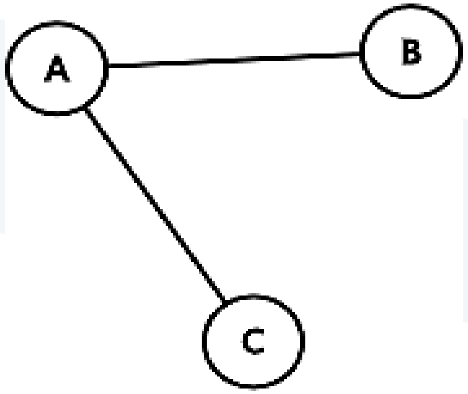
المراجع:

1- S. Russell, P. Norvig, Artificial Intelligence: A Modern Approach, Third Edition, April 2011, Artificial Intelligence, 175:935-937

2- Heuristics: intelligent search strategies for computer problem solving, Addison-Wesley Longman Publishing Co., Inc. 75 Arlington Street, Suite 300 Boston, MA, United States, ISBN:978-0-201-05594-8

محتويات المحاضرة

مراجعة لنظرية البيان
توسعة عقد الشجرة
أنماط البحث
البحث بدون معلومات
خوارزمية البحث بالعرض أولاً
مسائل



مراجعة لنظرية البيان:

سنقوم هنا بمراجعة سريعة لأهم الأفكار في نظرية البيان والتي ستلزمنا في هذا المقرر.

البيان هو عبارة عن مجموعتين:

مجموعة العقد:

وتعرف العقدة بالإنكليزية بالشكل vertex (جمعها vertices) كما تدعى node وتشبه مفهوم النقطة في الهندسة، تعرف باسم حالة state في خوارزميات البحث.

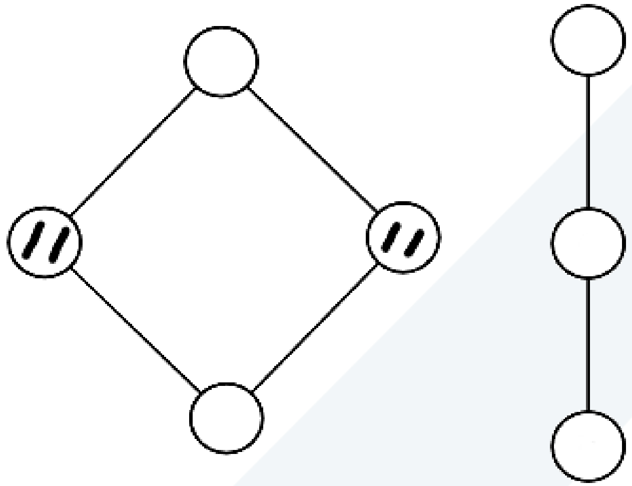
مجموعة الوصلات:

وتعرف الوصلة بالإنكليزية بالشكل edge كما تدعى قوس arc، وفي خوارزميات البحث تدعى أيضاً فِعل action.

في المثال جانباً: A,B,C هم عقد والوصلات هي AB, AC

نسمي النمط السابق من البيان **بياناً ذا صفات Labeled graph** حيث أن عقد البيان تملك أسماء (يمكن أيضاً أن تحوي العقد أو الوصلات على أوزان والتي تُعتبر أيضاً صفة).

يمكن في حالة **بيان لا يملك صفات** أن تتشابه عقدتين أو وصلتين بالخواص فندمجهم، ففي المثال التالي نرى أن العقدتين المشار لهما يرتبطان بنفس العقد فقمنا بدمجهم بعقدة واحدة.



البيان الموجه والبيان الغير موجه:

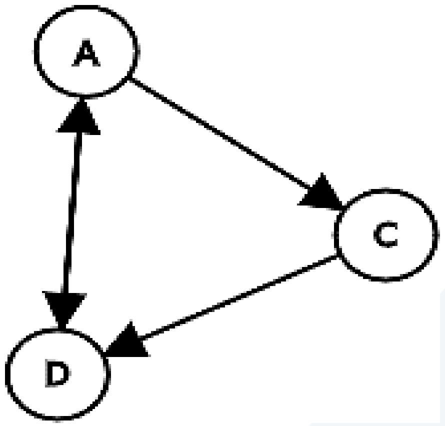
نسمي **البيان غير موجه** إذا كان من أجل أي وصلة ما يمكن الانتقال بالاتجاهين،

ففي المثال الأول يمكن الانتقال من A إلى C ومن C إلى A

بينما في **البيان الموجه** يتم تحديد اتجاه مسموح للانتقال، كما في المثال جانباً حيث الانتقال من C ل A غير ممكن.

ملاحظة:

عندما لا يتم وضع أسهم يتم اعتبار الانتقال بالجهتين.



البيان ذو الجذر:

يدعى بالإنكليزية Rooted graph وهو بيان يملك عقدة يمكن منها الوصول لبقية العقد،

ولا يمكن العودة فيه من عقدة لعقدة سابقة (لا يحوي حلقات)،

ندعو العقدة التي يمكن الوصول منها لبقية العقد بالجذر وتمثل في خوارزميات البحث حالة البداية لمسألة أو لعبة.

الشجرة:

تدعى بالإنكليزية Tree وهي بيان ذو جذر، يتميز بأنه من أجل أي عقدتين فيه هناك طريق واحد على الأكثر بين هاتين العقدتين، ولا تحوي الشجرة على أية حلقات.

ندعو العقدة التي تتفرع لعدة عقد بالأب Parent والعقد التي تتفرع لها بالأبناء Children،

نسمي أب الأب سلفاً Ancestor أو سابقاً Predecessor وابن الابن خلفاً Descendant أو تابعاً Successor والعقد من نفس المستوى أشقاء Siblings،

العقد التي لا تملك أبناء تعرف بالعقد الطرفية أو الأوراق،

عدد وصلات الشجرة مساوٍ لعدد العقد باستثناء عقدة: $|E| = |V| - 1$

في الشكل هنا، البيان اليساري هو بيان ذو جذر ولكنه ليس شجرة وذلك لوجود طريقين للعقدة D وهما ACD وABD.

بينما البيان اليميني هو شجرة،

بالنسبة للعقدة F، العقدة H هي عقدة شقيقة

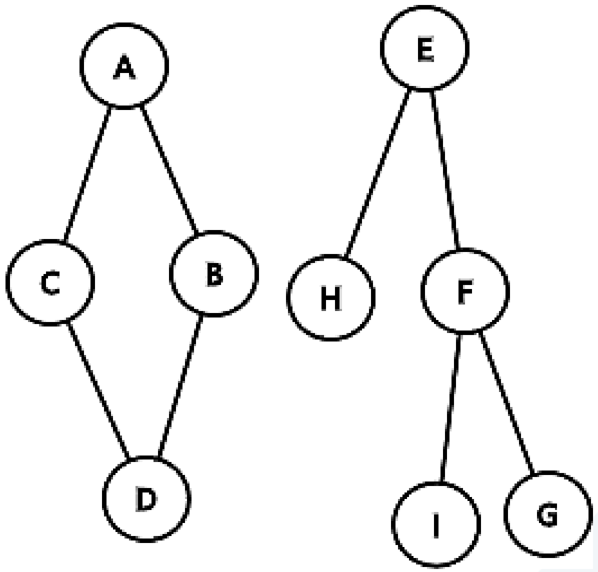
والعقدتان I و G هي أبنائها

والعقدة E هي أب العقدة F،

أيضاً العقدة E هي سلف العقدتين I G

والعقدتين I G هما خلف العقدة E،

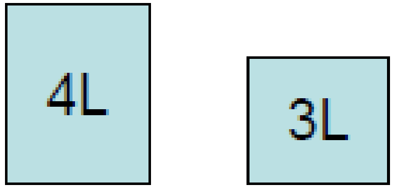
العقد I و G و H هي أوراق.



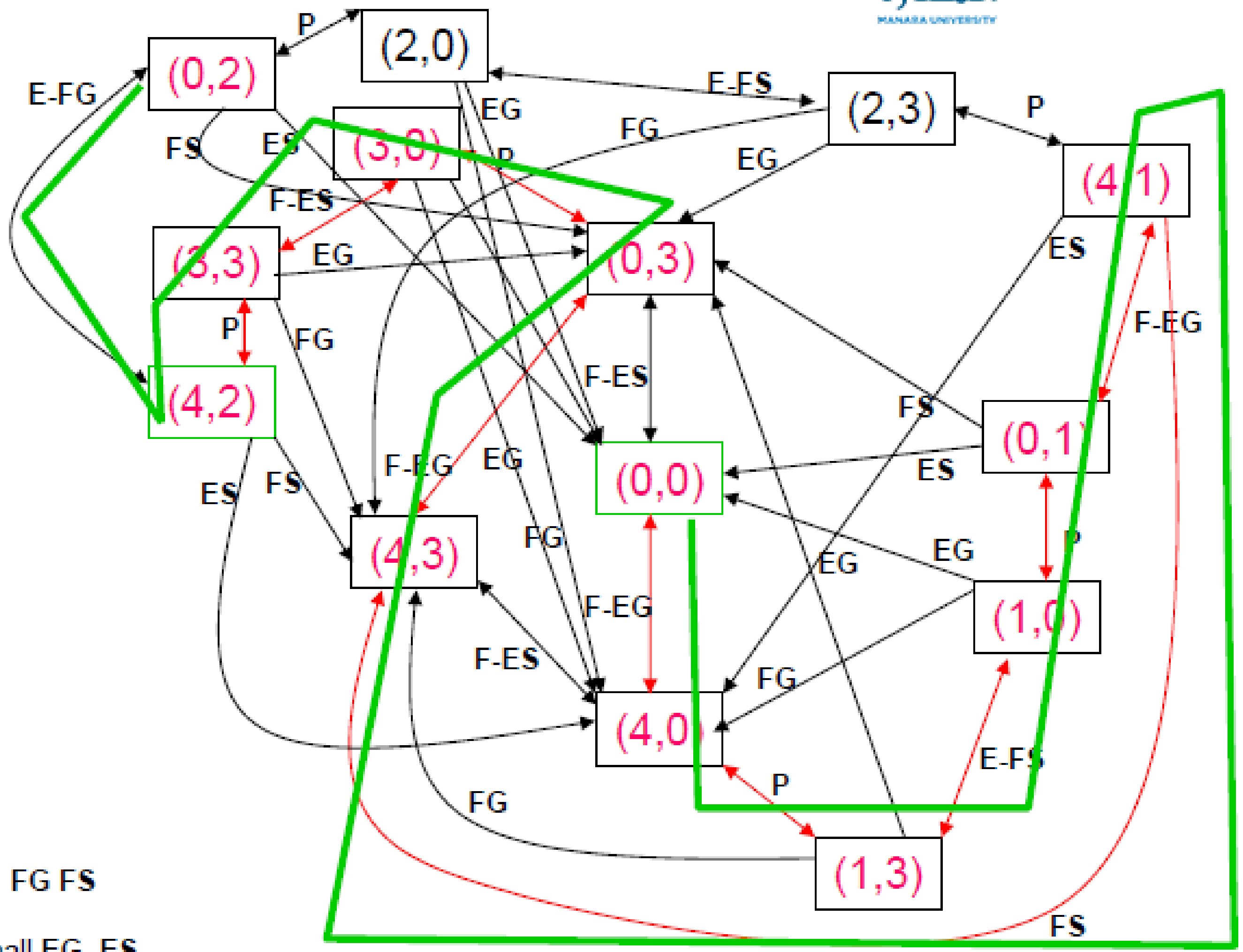
مسألة الوعاءين:

لدينا وعاءي ماء سعة الكبير منهما 4 لتر والصغير 3 لتر نريد ملء الكبير ب 2 لتر،

علماً أنه يمكننا فقط أن نعبئ الوعاء بأكمله أو نضع محتوى وعاء في وعاء آخر (في حال كانت سعته تسمح بذلك) أو أن نفرغ وعاء، صغ مسألة البحث.

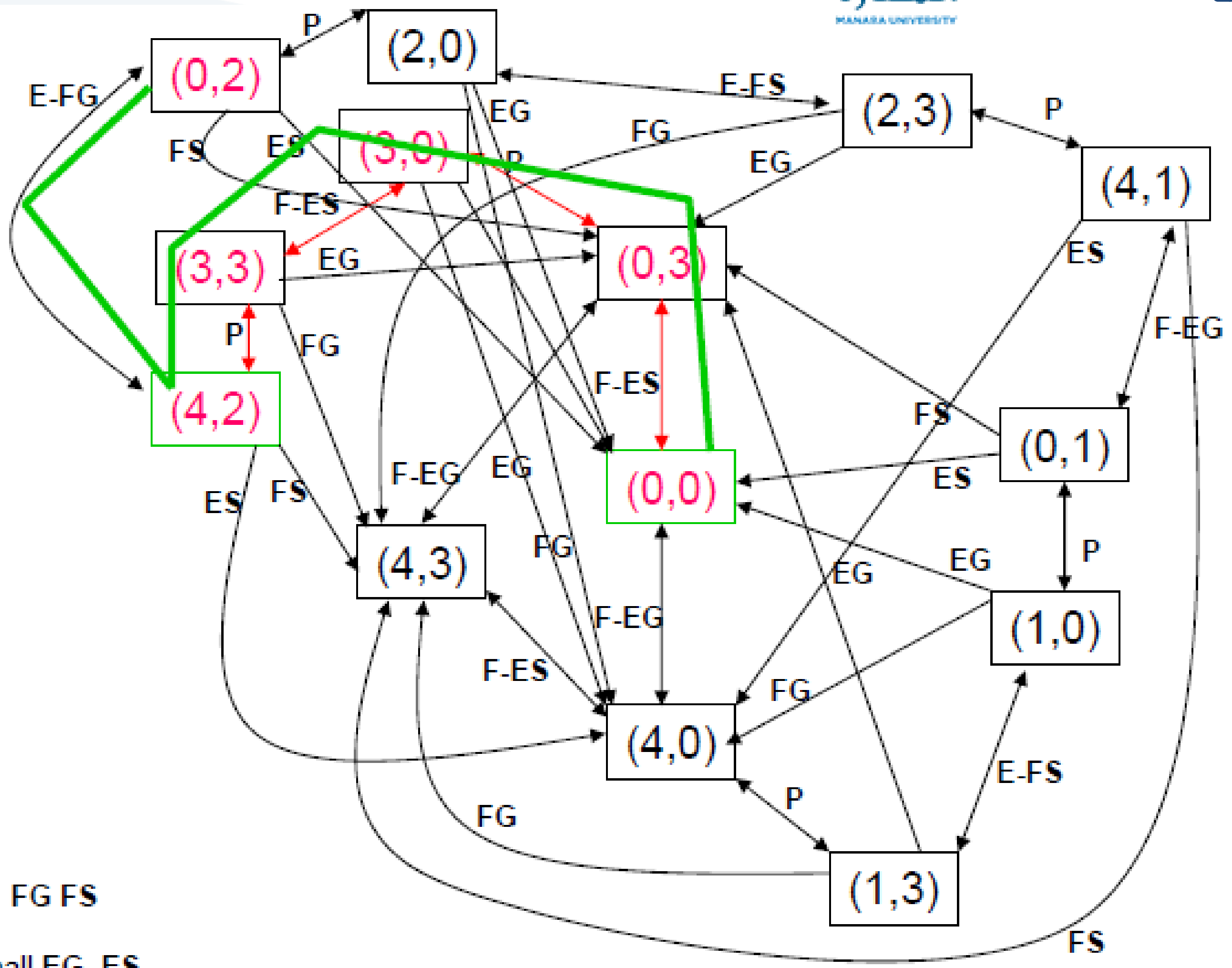


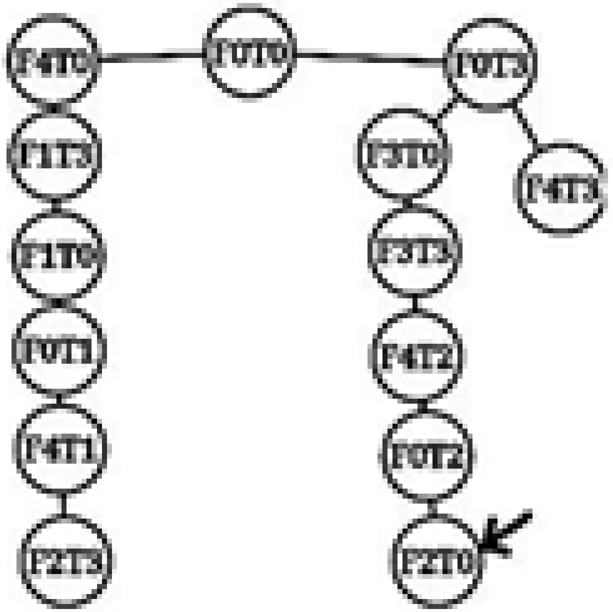
فضاء الحالات لمسألة الوعاءين: حل سيئ



Fill Grand/Small FG FS
Pour P
Emply Grand/small EG, ES
Empty and fill E-F or F-E

فضاء الحالات للمعضلة الوعائيين: حل جيد





Fill Grand/Small FG FS
Pour P
Emply Grand/small EG, ES
Empty and fill E-F or F-E

سنرمز هنا للوعاء الذي يحوي 4 لتر ب F والوعاء الذي يحوي 3 لتر ب T.

الحالة البدائية هي عندما يكون كلا الوعاءين فارغين F0T0.

<u>الانتقالات الممكنة هي:</u>

$$F0T0 \rightarrow (Fill\ F, F4T0)$$

قمنا بملء الوعاء الذي سعته 4 لتر.

$$F0T0 \rightarrow (Fill\ T, F0T3)$$

قمنا بملء الوعاء الذي سعته 4 لتر.

$$F4T0 \rightarrow (Put\ F\ in\ T, F1T3)$$

وضعنا محتوى الوعاء الذي سعته 4 لتر في الوعاء الذي سعته 3 لتر.

$$F4T0 \rightarrow (Empty\ F, F0T0)$$

قمنا بتفريغ الوعاء الذي سعته 4 لتر.

الهدف هو الحالة F2 T0،

ويمكن ألا تكون هناك كلفة أو أن تكون الكلفة هي الوصول بأقل عدد خطوات ممكن لتوفير كمية الماء المستهلكة

في هذه الحالة الحل الأفضلي F2T0 F0T2 F4T2 F3T3 F3T0 F0T3 F0T0،

- **كشرح مبسط للحل** نقوم بملء الوعاء الذي سعته 3 لتر ثم نُفرغ محتواه في الوعاء الذي سعته 4 لتر.
- بعدها نعاود ملء الوعاء الذي سعته 3 لتر ثم نضع محتوى الوعاء الذي سعته 3 لتر في الوعاء الذي سعته 4 لتر.
- وبما أن الوعاء الصغير يحوي 3 لتر والوعاء الكبير يحوي 3 لتر سيصبح الوعاء الصغير يحوي لترين والوعاء الكبير 4 لتر
- نفرغ محتوى الوعاء الكبير ونضع محتوى الوعاء الصغير في الكبير وبذلك نكون حصلنا على حل بأقل عدد من الخطوات.

توسعة عقد الشجرة:

نهدف لتحويل المسائل إلى بيان لأن الخوارزميات تتعامل مع البيان بشكل جيد كما أنه يصبح أسهل للاستيعاب،

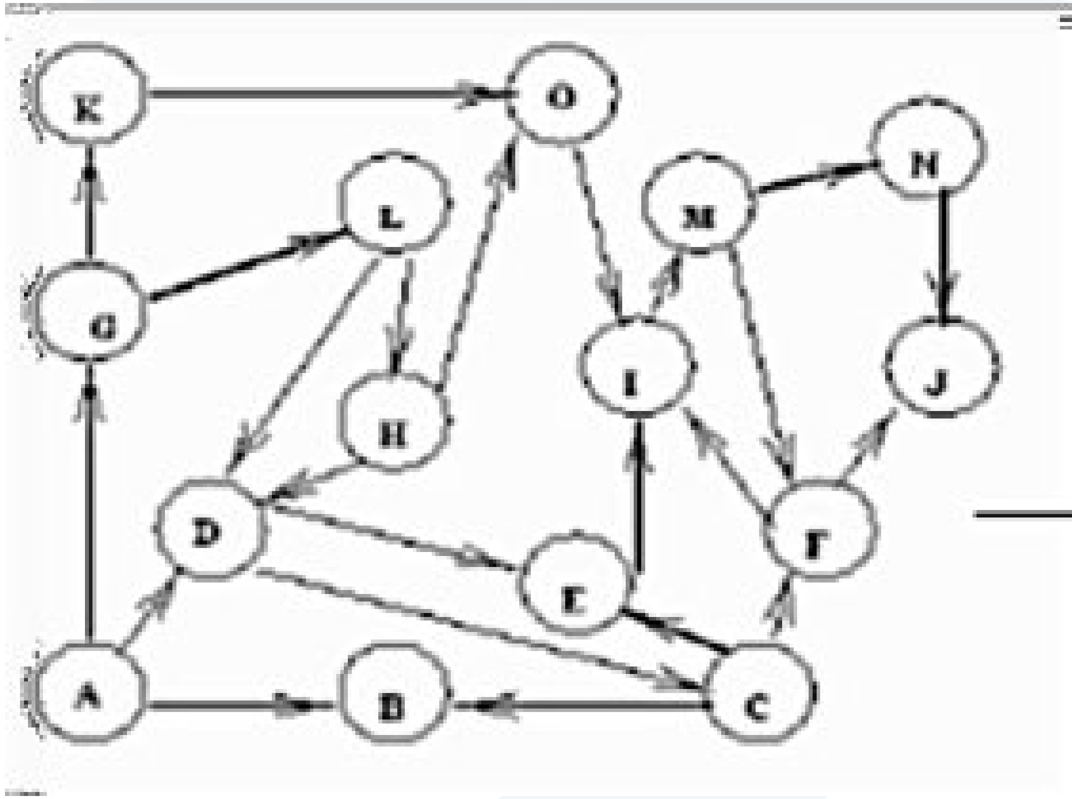
ونهدف بالضبط لتحويل البيان لشجرة، فعملية إضافة عقد أبناء **تعرف بالتوسعة** وتختلف خوارزميات البحث بكيفية القيام بهذه التوسعة، **أي أن خوارزمية البحث المستعملة تحدد كيفية توسعة الشجرة.**

أهم المشاكل التي تواجهنا في عملية توسيع الشجرة هي وجود حلقة أو عدة طرق مختلفة لنفس العقد وكيفية التعامل مع هذه الأمور وفق ما تحدده خوارزمية البحث،

ويلزمنا طريقة لمعرفة ما هي العقد التي مررنا بها لكي لا نمر بها مرتين فنضيف على شجرة البحث مجموعة تعرف بالمجموعة المُستكشَفة Explored set أو اللائحة المغلقة Closed List لكي لا نزور نفس العقدة مرتين.

- نسمي مجموعة الأوراق القابلة للتوسعة عند أي مرحلة **بالمجموعة الأمامية Frontier**،
- وبما أن خوارزميات البحث في البيان مثل DFS/BFS لا تقوم بتكرار العقد وهي خالية من الحلقات فيمكن أن نشبه العملية ببناء شجرة من بيان،
- حيث أن المجموعة الأمامية تقسم البيان للمجموعة المُستكشفة والمجموعة الغير مستكشفة وبالتالي مهما كان الطريق الذي ينطلق من الحالة الابتدائية فهو سيمر بالمجموعة الأمامية.

كيفية تحويل بيان إلى شجرة.



A problem

Search Tree for the Problem

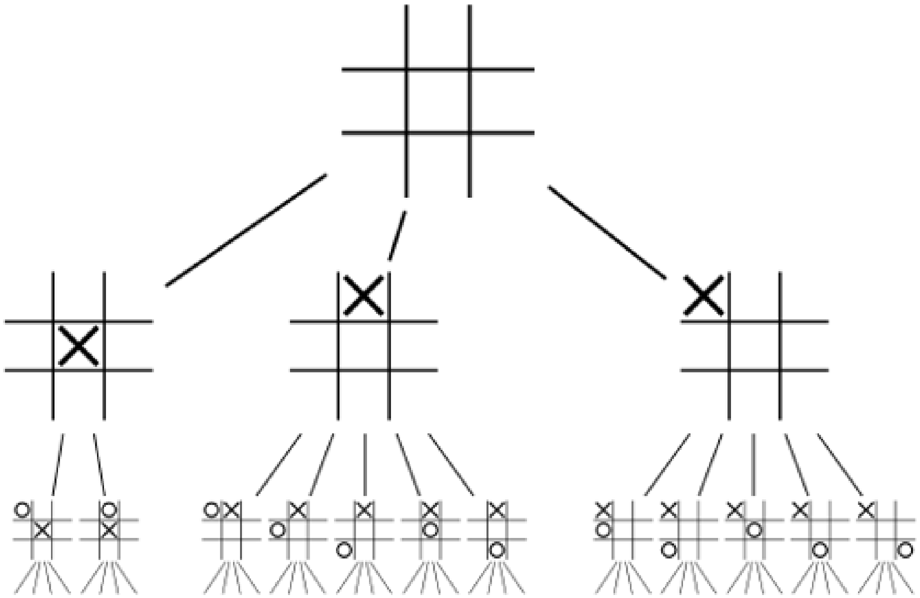
مثال:

في لعبة X-O الحد الأعلى للاحتمالات هو 9!

فالمستوى الأول من شجرة البحث يمثل ال X في المواقع المحتملة التسعة

والمستوى الثاني هو موقع ال O في المواقع الثمانية المتبقية حسب موقع ال X السابق

وتمثل الصورة جانباً جزءاً من شجرة الاحتمالات الممكنة والتي تنطلق من حالة البداية عندما تكون كل المربعات فارغة.



تمثيل المشكلة باستخدام فضاء الحالة:

يتم تمثيل فضاء الحالة باستخدام 4 عناصر بالشكل [N,A,S,GD] حيث:

N: هي مجموعة عقد أو حالات البيان.

A: هي مجموعة أضلاع أو وصلات البيان.

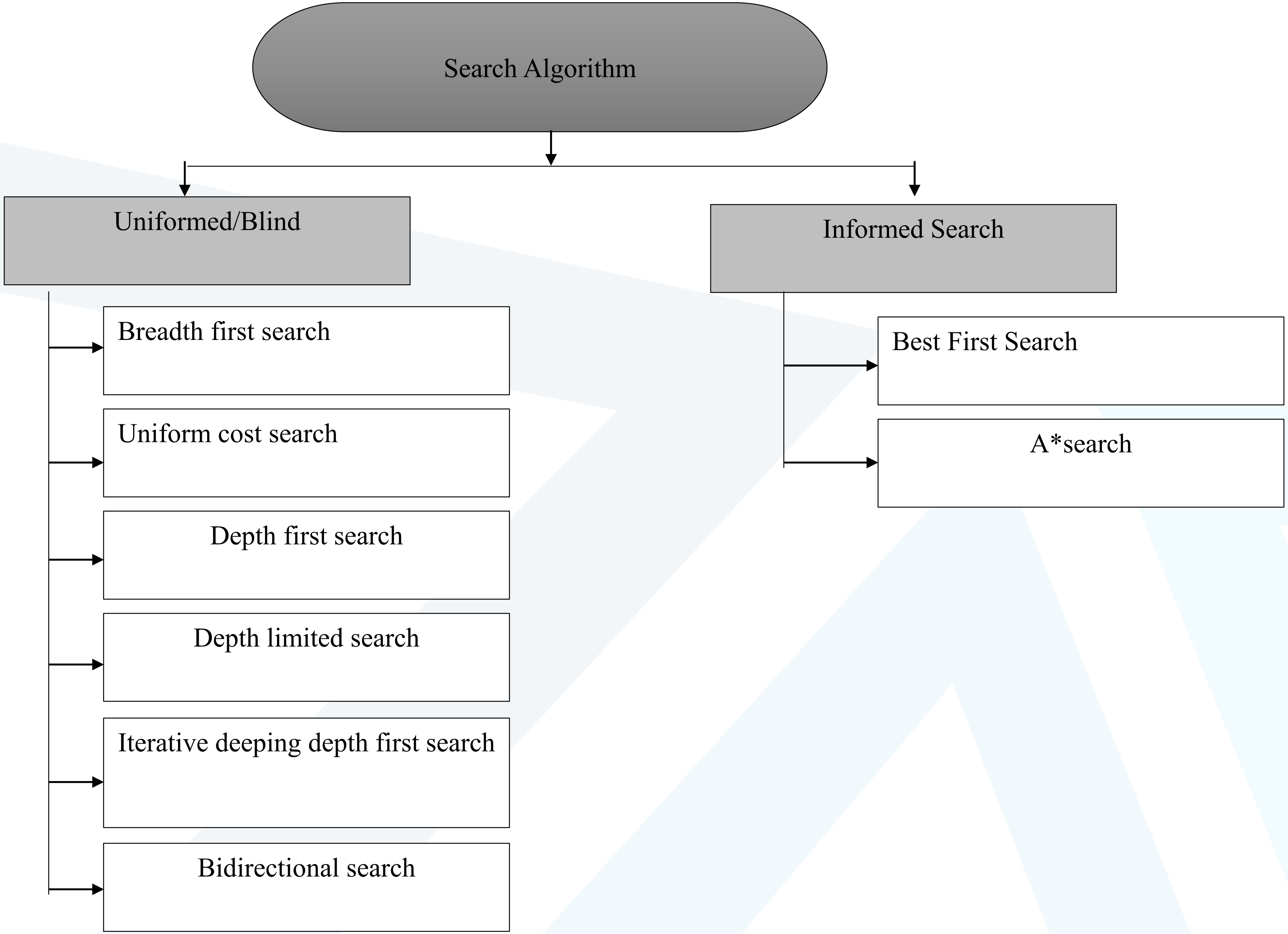
S: هي مجموعة جزئية غير خالية من N وتحتوي حالات البداية للمسألة.

GD: هي مجموعة جزئية غير خالية من N وتحتوي حالات النهاية للمسألة.

أنماط البحث:

وهم أربعة:

1. بحث بدون معلومات Uninformed Search،
2. بحث مع معلومات Informed Search،
3. تحقيق القيود Constraint satisfaction: ويهدف لإيجاد قيم لمجموعة من المتحولات،
4. بحث في بيئة تنافسية Adversarial search: ويُستعمل في الألعاب لإيجاد استراتيجيات فعالة للعب وربح الألعاب التي يلعبها لاعبين.


Search Algorithm
Uniformed/Blind
Informed Search
Breadth first search
Uniform cost search
Depth first search
Depth limited search
Iterative deeping depth first search
Bidirectional search
Best First Search
A*search

البحث بدون معلومات:

- ❑ كما رأينا سابقاً، تحدد المسألة البيان والهدف ولكنها لا تحدد الطريق الذي يجب اختياره من المجموعة الأمامية فهذه الأمور تحددها خوارزمية البحث،
- ❑ لا تمتلك خوارزميات البحث بدون معلومات أية معلومة عن الهدف باستثناء المعلومات الموجودة في المسألة، والطرق الممكنة تختلف فقط بالرتيب وطول الأفعال.
- ❖ فخوارزميات البحث بدون معلومات لا تملك أي نوع من المعلومات المتعلقة ببعد أو موقع الحالة الهدف وتتصرف بطريقة تشبه القوة العمياء Brute force،

- ولهذا تعرف بخوارزميات القوة العمياء أو خوارزميات البحث الأعمى،
- وهي طريقة حدسية وطبيعية ومباشرة يتم بها تجربة كل الاحتمالات الممكنة
- هناك العديد من المسائل اليومية التي قد نستعمل بها القوة العمياء مثل إيجاد أقصر طريق لمكان ما
- وبالواقع العديد من المسائل تحل باستخدام القوة العمياء بالرغم من وجود خوارزميات تعطي حلاً أفضلياً.

لتوضيح مفهوم القوة العمياء لنعتبر أننا نقوم بلعب لعبة حبل المشنقة،

عند اختيار حرف خاطئ يتم إزالة الحرف ويقوم اللاعب بالابتعاد عن الحرف وتجربة أحرف أخرى،

أي أن المعلومات التي تشير لعدم كون الحرف محتملاً تعني أن الطرق التي عقدها تحوي هذا الحرف هي طرق تقود لنهاية مغلقة ولا تقود للحل،

أي أن المعلومات التي قد نحصل عليها عند بداية اللعبة أو تطورها تؤدي لتقليل الاحتمالات وقد تزيل أفرع كاملة من الشجرة فسيكون تجربة كلمات مثل مبهر، مبارك... نوعاً من السخافة إن عرفنا أن حرف الباء غير موجود،

ولكن خوارزمية القوة العمياء حتى إن ظهرت معلومة تقول أن حرف الباء غير موجود ستقوم بتجريب كلمات مثل مبهر، ولهذا تُعرف أيضاً بخوارزميات البحث دون معلومات.

بعض الأمثلة على خوارزميات البحث دون معلومات:

البحث بالعرض أولاً BFS

البحث بالعمق أولاً DFS

البحث المحدود بالعمق DLS

البحث بالعمق أولاً المُعمَق بشكل تكراري IDDFS

البحث ثنائي الاتجاه BS

البحث ذو التكلفة الموحدة UCS

Breadth first search
Depth first search
Depth limited search
Iterative deeping depth first search
Bidirectional search
Uniform cost search

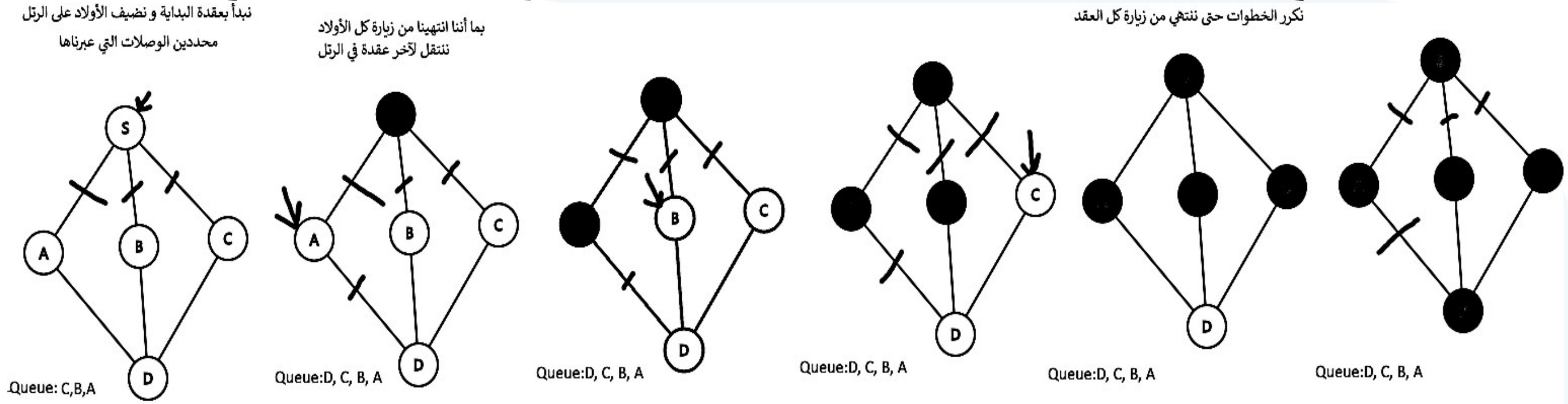
خوارزميات البحث بالعرض أولاً هي كما يلي:

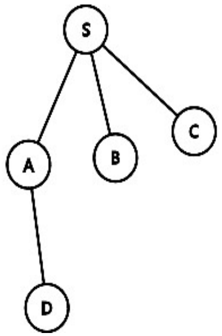
```
begin                                   %initialize
open:= [Start];
closed:= [];
while open ≠ [] do                 %states remaining
 begin
   remove leftmost state from open, call it
   X;
   If X is a goal then return SUCCESS %goal
found
   else begin
    generate children of X;
    put X on closed;
    discard children of X if already on
open or closed;                       %loop check
    put remaining children on right end of
open                               %queue

 end
  end;
 return FAIL
end.
                        %no states left
```

لتذكر البحث بالعرض أولاً سنحل المسألتين التاليتين:

مسألة 1:

عند القيام بالبحث بالعرض أولاً Breadth First Search ننطلق من عقدة البداية ونضيف العقد الأولاد من **اليسار لليمين** ثم نضعهم في الرتل Queue والذي يتبع إستراتيجية أول من دخل هو أول من يخرج FIFO، توضح الصورة الخطوات:



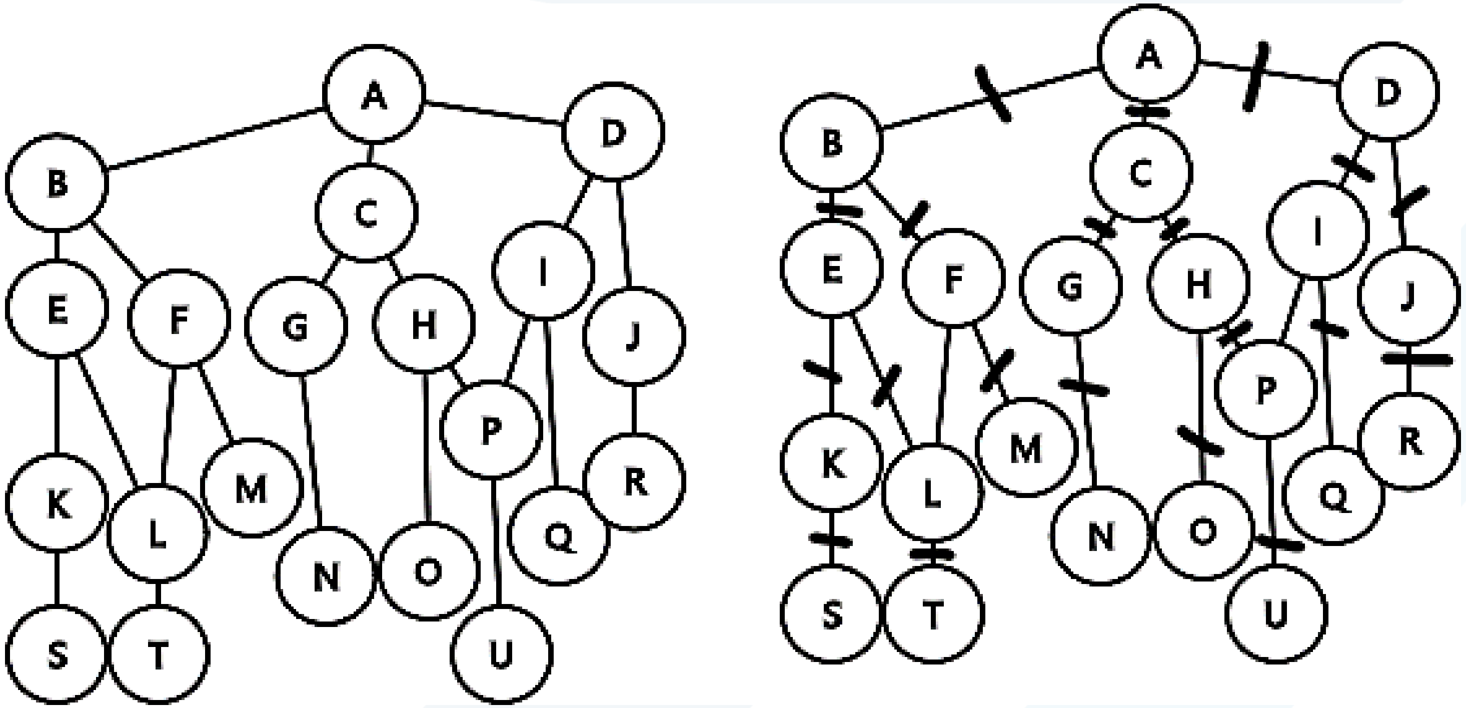


يجب الإشارة إلى عدم معاودة العقد الموضوعة في الرتل (كما العقدة D في المثال الثالث).

ملاحظة: البحث حسب العرض يقوم بعبور الشجرة حسب المستويات.

مسألة 2:

ليكن لدينا البيان الموضح في الصورة جانباً المطلوب تطبيق البحث بالعرض أولاً عليه.



هناك طريقة أخرى للتعبير عن خطوات الخوارزمية ويفضل استعمالها

نقسم العقد لمجموعتين Open و Closed العقد في المجموعة Open هي العقد التي نتعامل معها والعقد في المجموعة Closed هي العقد التي انتهينا منها. ففي بداية المسألة نبدأ بالعقدة A

Open = [A],Closed = []

وبحكم أننا لم ننتهي من أي عقدة فالمجموعة الثانية فارغة،

ثم نزيل A من المجموعة Open ونضيفها للمجموعة Closed ونضع في المجموعة Open أبناء A.

Open = [B,C,D],Closed = [A]

وهكذا .... نكرر الخطوات.

Open = [C,D,E,F],Closed = [B,A]

Open = [D,E,F,G,H],Closed = [C,B,A]

Open = [E,F,G,H,I,J],Closed = [D,C,B,A]

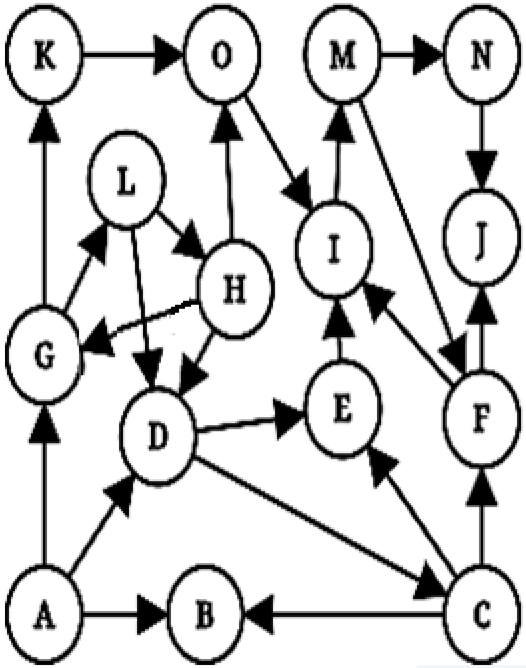
Open = [F,G,H,I,J,K,L],Closed = [E,D,C,B,A]

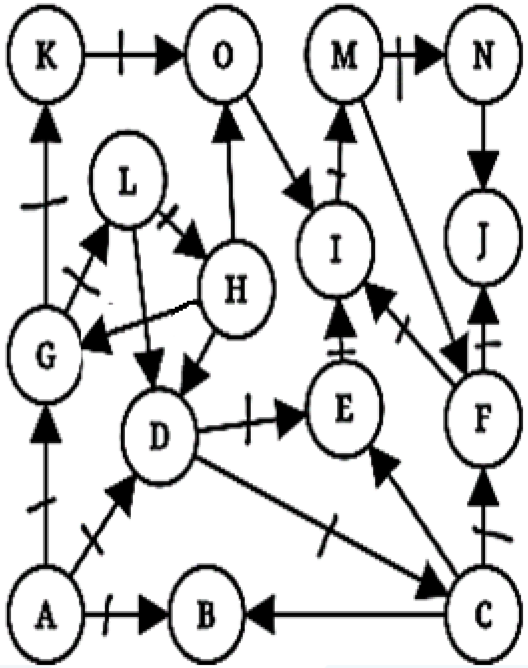
Open = [G,H,I,J,K,L,M],Closed = [F,E,D,C,B,A]

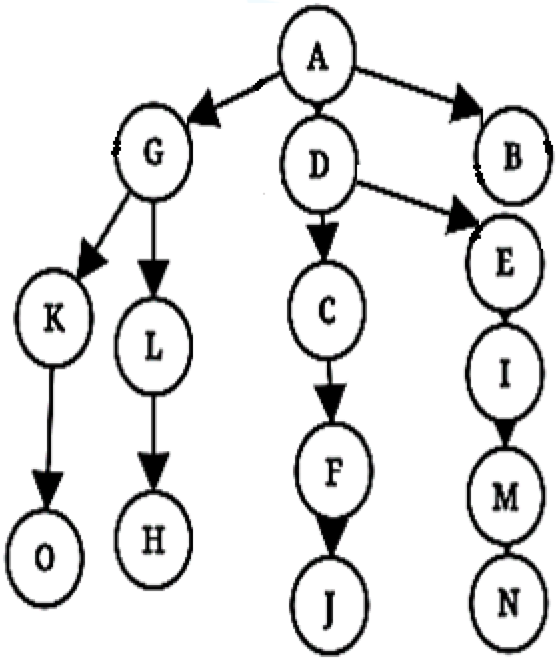
لا نضيف العقدة L مرتين لأنها موجودة في Open (وإن كانت في Closed نقوم بنفس الشيء) ونكمل حتى تصبح كل العقد إما Open أو Closed.

مسألة:

ليكن لدينا البيان في الصورة جانباً والمطلوب إيجاد شجرة البحث باستخدام البحث بالعرض أولاً انطلاقاً من العقدة A.







سنبدأ من العقدة A كما وموجود في نص المسألة.

Open = [A],Closed = []

ثم نضيف أبناء A ونشير للأضلاع التي عبرناها.

Open = [G,D,B],Closed = [A]

ونكرر الخطوات كما في المسألة السابقة فنضيف أبناء G ثم أبناء D و وهكذا ....

Open = [D,B,K,L],Closed = [G,A]

Open = [B,K,L,E,C],Closed = [D,G,A]

Open = [K,L,E,C],Closed = [B,D,G,A]

Open = [L,E,C,O],Closed = [K,D,G,A]

Open = [E,C,O,H],Closed = [L,K,D,G,A]

Open = [C,O,H,I],Closed = [E,L,K,D,G,A]

تتمة:

Open = [O,H,I,F],Closed = [C,E,L,K,D,G,A]

Open = [H,I,F],Closed = [O,C,E,L,K,D,G,A]

Open = [I,F],Closed = [H,O,C,E,L,K,D,G,A]

Open = [F,M],Closed = [I,H,O,C,E,L,K,D,G,A]

Open = [M,J],Closed = [F,H,O,C,E,L,K,D,G,A]

Open = [J,N],Closed = [M,F,H,O,C,E,L,K,D,G,A]

Open = [N],Closed = [J,M,F,H,O,C,E,L,K,D,G,A]

Open = [],Closed = [J,M,F,H,O,C,E,L,K,D,G,A]

بما أن جميع العقد أصبحت Closed فهذا يعني أننا انتهينا من خوارزمية البحث حسب العرض أولاً، فكل ما يتبقى أن نحافظ على الأضلاع التي أشرنا لها ونمحي بقية الأضلاع الغير مشارة، وبالتالي تتولد شجرة البحث كما في الشكل اليميني في الصورة أعلاه.

ملاحظة:

عند تطبيق البحث حسب العرض أولاً نبدأ من اليسار لليمين في كل مستوى.