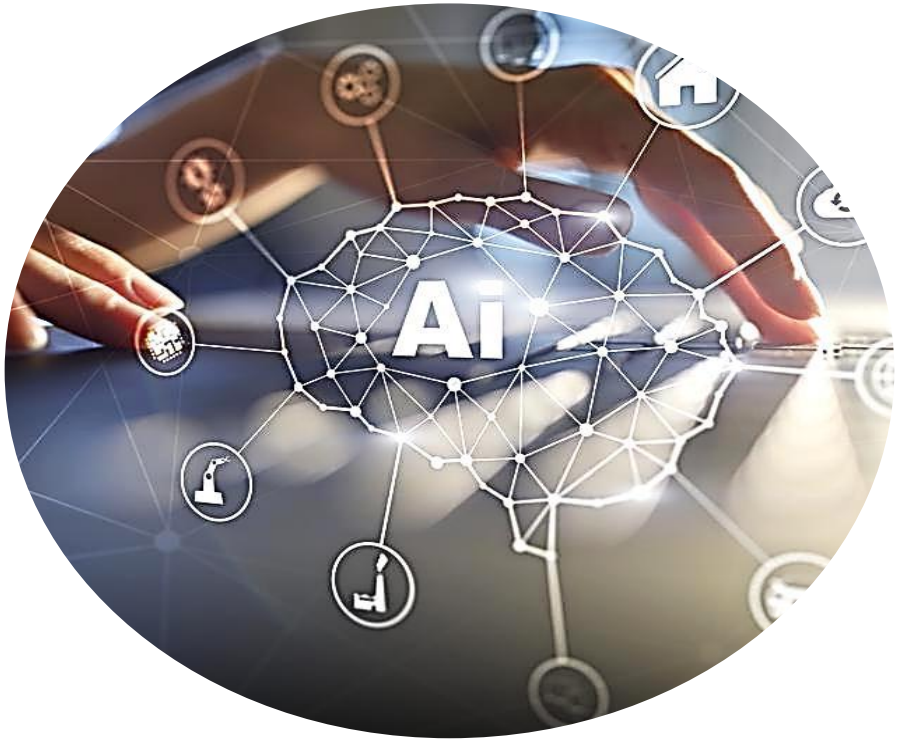
مقدمة في الذكاء الصنعي

الهندسة المعلوماتية



مدرس المقرر

د.بلال شيحا

المحاضرة الرابعة

البحث بالعرض أولا
Breadth First Search(BFS)

In breadth-first search

- explores the space in a level-by-level fashion
- only when there are no more states to be explored at a given level does the algorithm move on to the next level

في البحث بالعرض أولا

- يستكشف الفضاء بطريقة مستوى بمستوى
- فقط عندما لا يكون هناك المزيد من الحالات ليتم استكشافها على مستوى معين ، تنتقل الخوارزمية إلى المستوى التالي

في هذه الخوارزمية : توضع العقد المولدة في رتل FIFO الداخل أولا يخرج أو لا

البحث بالعرض أولا
Breadth First Search(BFS)

مثال لشجرة البحث بالعرض أولا Breadth–first search tree

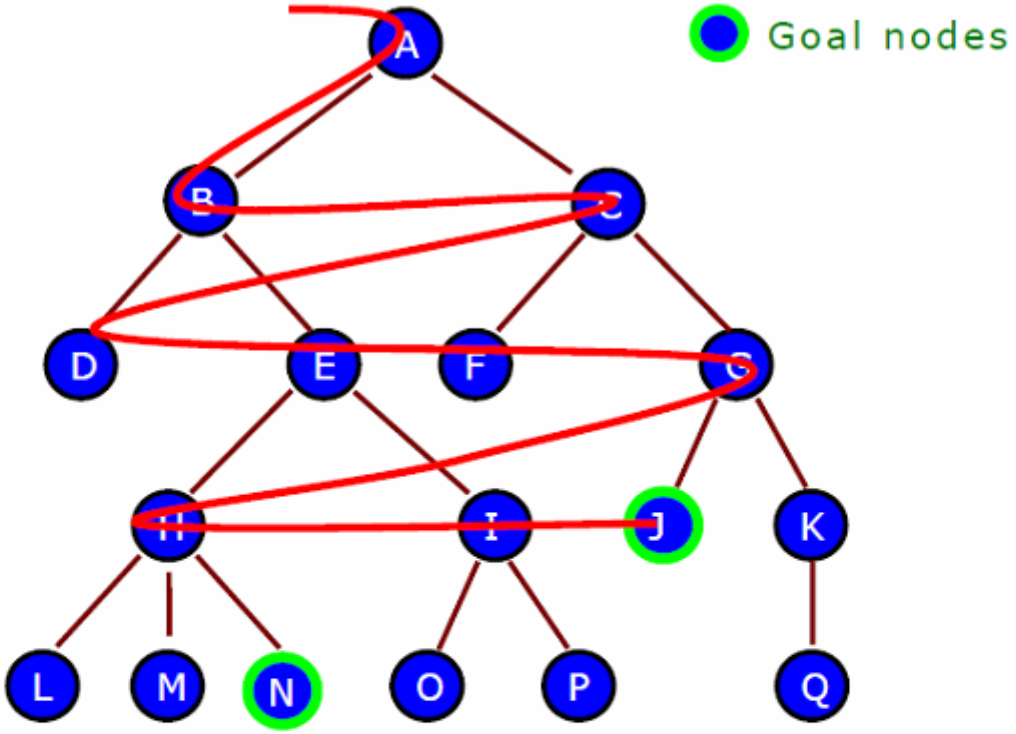


Fig. Breadth-first search (BFS)

لاحظ أن النقاط المستكشفة وفق البحث بالعرض هي:

A B C D E F G H I J K L M N O P Q

بعد البحث في النقطة A ثم B ثم C، يستمر البحث إلى المستوى الثاني في النقطة D ثم E ...حتى نهاية المستوى الثاني، ثم ينزل إلى المستوى الثالث مادامت النقطة الهدف لم تستكشف بعد، ونلاحظ من الشجرة أعلاه أن النقطة الهدف J قد اكتشفت قبل الوصول للنقطة الهدف N .

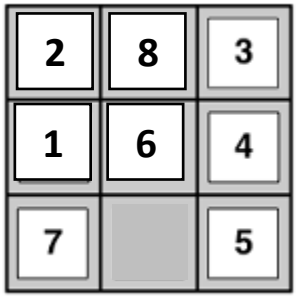
البحث بالعرض أولاً

```
function breadth_first_search;

begin
   open := [Start];                                          % initialize
   closed := [ ];
   while open ≠ [ ] do                                       % states remain
      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                                               % no states left
end.
```

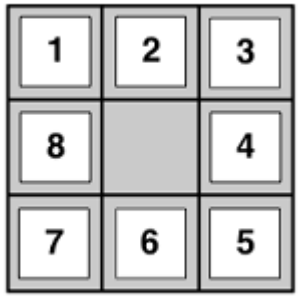
مسألة أحجية القطع الثمانية
8-puzzle

- المطلوب الوصول إلى الحالة الهدف انطلاقاً من حالة ابتدائية ما.



Start State

حالة ابتدائية



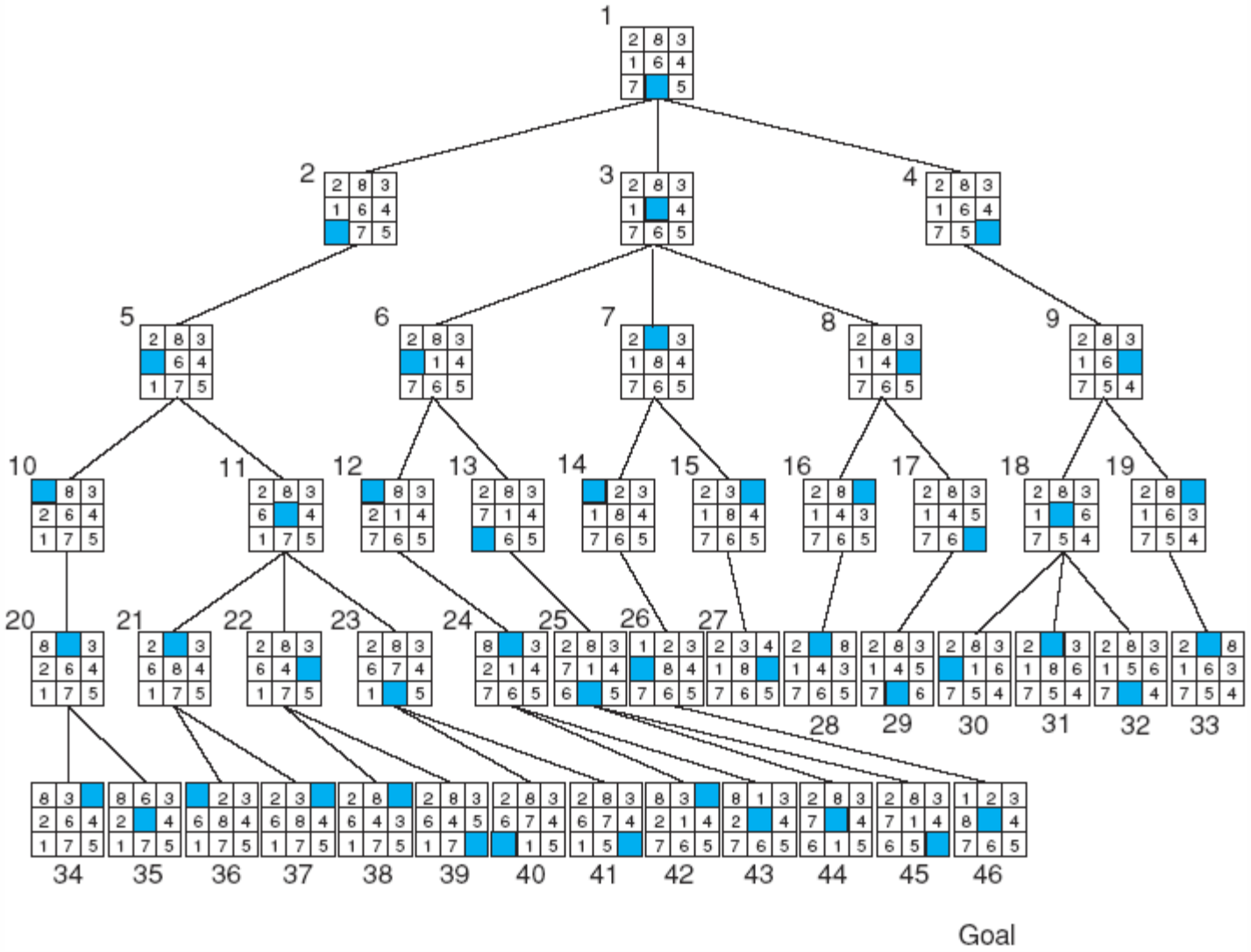
Goal State

الحالة الهدف

مسألة أحجية القطع الثمانية
8-puzzle

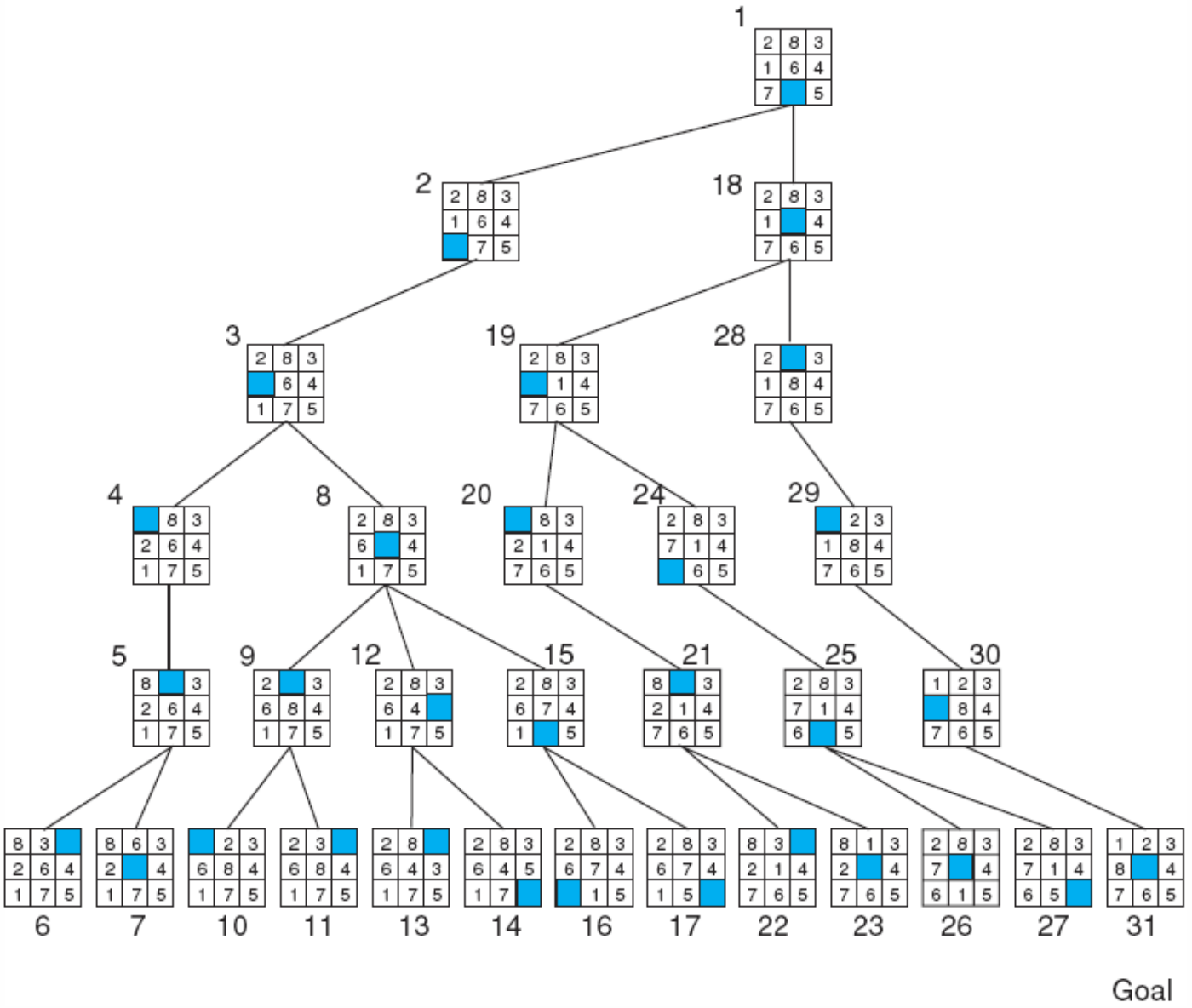
يوضح الشكل الحالات التي تمت إزالتها من قائمة الفتح وفحصها في البحث بالعرض أولا لبيان **8-puzzle** . تتوافق الأقواس مع تحركات الفراغ للأعلى و لليمين و للأسفل و لليسار . يشير الرقم الموجود بجوار كل حالة إلى الترتيب الذي تمت إزالته به من قائمة الفتح . لا تظهر الحالات التي تُركت مفتوحة عند توقف الخوارزمية.

مثال البحث بالعرض أولاً



يعطي الشكل التالي بحثًا بالعمق أولًا عن **(8-puzzle)** .كما في السابق، يتم إنشاء الفضاء بواسطة قواعد "الانتقال إلى الفراغ" الأربعة (للأعلى و للأسفل ولليسار ولليمين). تشير الأرقام الموجودة بجانب الحالات إلى الترتيب الذي تم تقييمها به، أي إزالتها من قائمة الفتح . لا يتم عرض الحالات التي تُركت مفتوحة عند العثور على الهدف . تم فرض حد عمق 5 على هذا البحث لمنعه من الضياع في أعماق الفضاء.

مثال البحث في العمق أولاً بعمق بحث = 5



مقارنة بين البحث بالعرض أولا وبالعمق أولا

يعتمد اختيار "العمق أولاً" أو "العرض أولا" ً على المشكلة المحددة التي يتم حلها.

- تشمل الميزات الهامة أهمية العثور على أقصر طريق إلى الهدف ، وعامل التفرع ضمن الفضاء ، وموارد حساب الوقت والمكان المتاحة ، ومتوسط طول المسارات إلى عقدة الهدف ، وما إذا كنا نريد جميع الحلول أو الحل الأول فقط
- عند اتخاذ هذه القرارات ، هناك مزايا وعيوب لكل نهج

مقارنة بين البحث بالعرض أولا وبالعمق أولا

بالعرض أولا

- يختبر دائمًا جميع العقد في المستوى n قبل المتابعة إلى المستوى n + 1 ، يبحث البحث بالعرض أولا دائمًا عن أقصر مسار إلى عقدة الهدف.
- في حال كان هناك حل بسيط للمشكلة ، يتم إيجاد هذا الحل.
- لسوء الحظ ، إذا كان هناك عامل تفريع سيئ ، أي أن الحالة لديها متوسط عدد مرتفع من الأحفاد ، فقد يمنع الانفجار التوافقي الخوارزمية من إيجاد حل باستخدام الفضاء المتاح و يرجع هذا إلى حقيقة أن جميع العقد لكل مستوى من مستويات البحث يجب أن تظل مفتوحة.
- بالنسبة لعمليات البحث العميقة ، أو فضاء الحالة ذي عامل التفرع العالي ، يمكن أن يصبح هذا مرهقًا للغاية.
- يعد استخدام فضاء البحث بالعرض أولاً ، المقاس من حيث عدد الحالات المفتوحة ، دالة أسية لطول المسار في أي وقت
- إذا كان لكل حالة متوسط عدد الأبناء B ، فهناك B^n states في المستوى

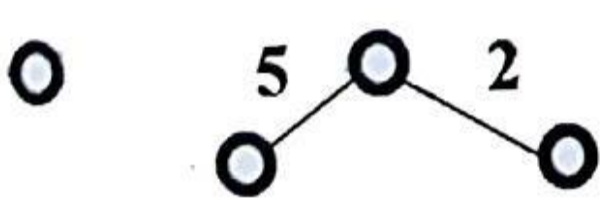
مقارنة بين البحث بالعرض أولا وبالعمق أولا

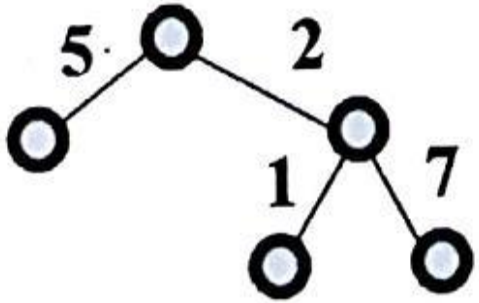
بالعمق أولا

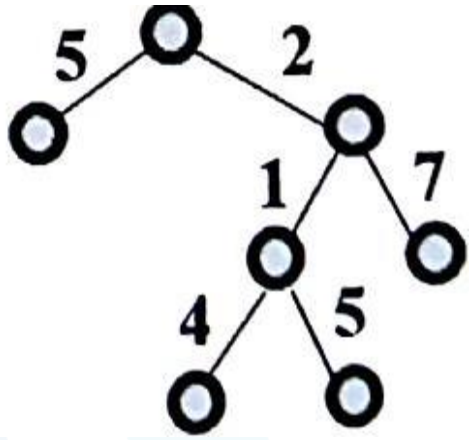
- يبدأ البحث في العمق أولاً في الوصول بسرعة إلى فضاء بحث عميق.
- إذا علمت أن مسار الحل سيكون طويلاً، فلن يضيع البحث بالعمق أولًا الوقت في البحث عن عدد كبير من الحالات "السطحية" في البيان.
- يمكن أن "يضيع" بحث العمق أولاً في عمق البيان، أو يفقد المسارات الأقصر للوصول إلى الهدف أو حتى أن يعلق في مسار طويل بلا حدود .
- إنها أكثر فاعلية لفضاء البحث الذي يحتوي على العديد من الفروع لأنه لا يتعين عليها الاحتفاظ بجميع العقد عند مستوى معين في القائمة المفتوحة.
- استخدام الفضاء للبحث في العمق أولاً هو دالة خطية لطول المسار إذا كان لكل حالة متوسط عدد الأبناء B لكل حالة، فإن هذا يتطلب استخدامًا إجماليًا لفضاء B x n states للانتقال إلى مستويات n في عمق الفضاء.

البحث بالكلفة الموحدة أولاً
Uniformed Cost –first Search

تعتمد هذه الخوارزمية على مبدأ تطوير العقدة ذات الكلفة الأقل و بالتالي ان أي عقدة هدف نصل اليها تكون الحل الأمثل.







خوارزمية البحث بالكلفة الموحدة أولاً: هنا ننتقل للعقدة ذات الكلفة التراكميّة الأقل مع ملاحظة أننا نقارن بكلف أوراق الشجرة كلها وليس المستوى الذي وصلنا له فقط أي أننا نستخدم رتل أفضلية (تطوير لخوارزمية البحث بالعرض أولا من خلال تحسين الرتل ليصبح رتل مع أفضلية)

خواص خوارزمية البحث بالكلفة الموحدة أولاً
Uniformed Cost-First Search

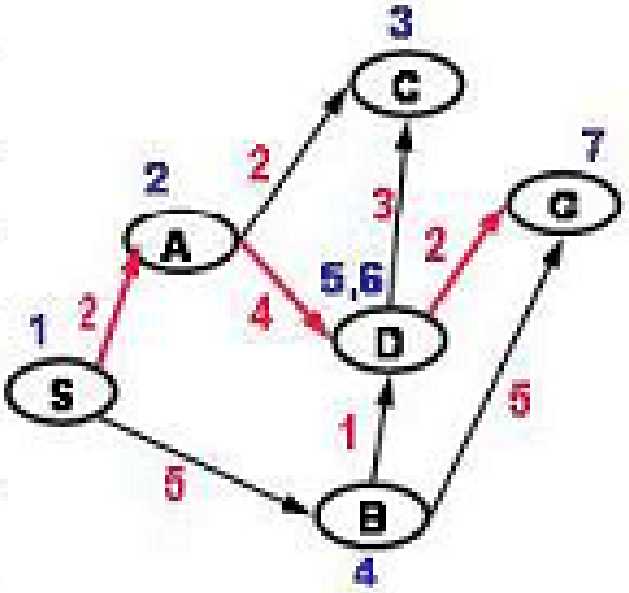
بشكل عام تقوم بزيارة العقد التي لم تتم زيارتها ذات الكلفة الأقل . و يمكن تحقيق هذه المنهجية عن طريق تخزين العقد المكتشفة و التي لم تتم زيارتها ضمن Priority queue الرتل ذو الأولوية وتعتمد فكرة الأولوية هنا على كلفة الطريق (الأصغر إلى الأكبر).

ملاحظة :

بسبب أهمية كلفة الطريق في هذه المنهجية لا نقوم بتطبيق مفهوم visited كما في الخوارزميتين السابقتين بل سيتم تخزين الطريق الأقصر وعند فحص المسارات المتبقية ضمن فضاء الحلول اذا كان هناك طريق آخر أقصر من الطريق المخزن يتم الاحتفاظ به وإهمال الحل القديم (الطريق السابق.)

البحث بالكلفة الموحدة أولاً
Uniform Cost - first Search

	Q
1	(0 S)
2	(2 A S) (5 B S)
3	(4 C A S) (6 D A S) (5 B S)
4	(6 D A S) (5 B S)
5	(6 D B S) (10 G B S) (6 D A S)
6	(8 G D B S) (9 C D B S) (10 G B S) (6 D A S)
7	(8 G D A S) (9 C D A S) (8 G D B S) (9 C D B S) (10 G B S)



Note:

Expand the cheapest unexpanded node

Implementation :

frontier = priority queue ordered by path cost g (n)

معايير تقويم عملية البحث

- الشمولية completeness والتي تعطينا فيما إذا كانت الاستراتيجية المستخدمة تضمن الوصول إلى حل، إذا كان هنالك من حل. و ننطلق من عمق بحث ومعامل تفريع محددين وكلفة أصغرية للوحدة في حال كان عمق البحث لانهائياً.
- التعقيد الزمني time complexity كم نستغرق للوصول إلى حل في الحالة الأسوأ ويرتبط ذلك بعدد العقد الموسعة أو المنشورة.
- التعقيد المكاني space complexity يحدد بحجم الذاكرة التي نستغلها في الحالة الأسوأ، ويرتبط بالعدد الأعظمي للعقد في الذاكرة.
- الأمثلية optimality: هل نصل بشكل دائم إلى الحل ذي الكلفة الأقل (العمق والممر الأفضل) .

التعقيد الزمني والفراغي

يقاس التعقيد الزمني والفراغي من خلال:

- b: معامل التفريع الأعظمي لشجرة البحث (عدد العقد الأبناء الأعظمي انطلاقاً من العقدة التي تم توليدها).
- d: عمق الحل الأقل كلفة.
- m: العمق الأعظمي لفضاء الحالة (قد يكون لانهائي).

خواص خوارزمية البحث بالعمق أولاً
DFS

1- Complete :

- إن خوارزمية DFS مكتملة و منتهية .

2- Optimality :

- كما لاحظنا أن خوارزمية DFS تهدف إلى إيجاد طرق من حالة البداية إلى حالة النهاية ولكن ليس بالضرورة حل أمثلي .

3-Time :

- يمكن التعبير عن الكلفة الزمنية b^m حيث b هو معامل التفرع و m هي عمق شجرة البحث.

4- Space :

- يمكن التعبير عن الكلفة المكانية بما يلي : $b*m$

خواص خوارزمية البحث بالعرض أولاً
BFS

1- Complete :

- هذه الخوارزمية مكتملة كسابقتها

2- Optimality :

- تضمن خوارزمية البحث بالعرض إيجاد حل أمثلي

3- Time :

- يمكن التعبير عن الكلفة الزمنية b^{d+1} حيث تعبر b عن عامل التفرع و تعبر d عن عمق الحل في شجرة البحث.

4- Space :

- يمكن التعبير عن الكلفة المكانية b^{d+1}

خواص خوارزميةالبحث بالكلفة الموحدة أولاً
Uniformed Cost - First Search

1- Complete :

- هذه الخوارزمية مكتملة كسابقتها وذلك في حال كانت الأوزان أكبر أو تساوي الصفر.

2- Optimality :

- تضمن هذه الخوارزمية (UCS) إيجاد حل أمثلي.

3- Time :

- يمكن التعبير عن الكلفة الزمنية $O(b^{c/\epsilon})$ حيث c هي كلفة الحل الأمثل و ε وسطي كلفة الخطوة.

4- Space :

- يمكن التعبير عن الكلفة المكانية $O(b^{c/\epsilon})$.

البحث المعرفي
Informed Search

- **البحث المعرفي :** يعني اتمام البحث باستخدام معارف تخص المسألة قيد البحث

* خوارزميات البحث الأعمى لا تستخدم أي معلومات تخص هيكلية شجرة البحث من أجل تحسين كفاءة البحث و هي تقوم بالبحث ضمن فضاء الحالة حتى الوصول لحل.

* و بالتالي فإن حل المعضلات الحقيقية اعتمادا على هذه الخوارزميات سيؤدي إلى انفجار حسابي (عدد الحالات الممكنة كبير).

* تستخدم خوارزميات البحث الحدسي Heuristic كل المعلومات المتاحة لجعل عملية البحث أكثر كفاءة.

* المعلومات الحدسية Heuristic هي قاعدة أو طريقة تؤدي عموما إلى تحسين كفاءة عملية البحث.

Informed Search

- **يعتمد البحث المعرفي على حساب دالة التقويم**
- و يقيس تابع التقويم البعد عن الهدف.
- يتوجب بعدها اختيار العقدة التي تحصل على أفضل تقويم بما يوحي بأنها العقدة التالية الأفضل في السعي للوصول إلى الهدف.
- يتم تحقيق ذلك من خلال استخدامنا لقائمة نرتب بها العقد التالية ترتيبا تنازليا وفقا لدرجات أفضلياتها.
- و تم دراسة دوال تقويم مطورة وفقا لتقنيتي البحث بأفضل أولا (الجشع) best-first search(greedy) و البحث وفقا لخوارزمية A^* search.
- تعتبر الخبرة , البساطة و التوقع المدروس من العوامل التي تقلل من حيز البحث في فضاء الحلول.

Informed Search

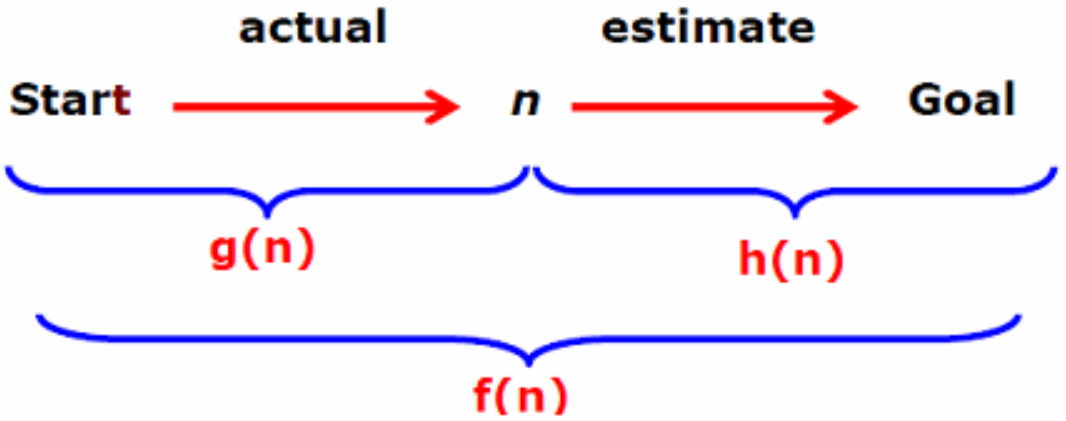
- نستخدم بعض التعاريف و التوابع التي تعرف بحدود المسألة المطروحة مثل:
- تابع التقويم evaluation function ***f(n)*** التابع الذي يحدد لنا نظريا الخطوة التالية. و يعطى بالعلاقة

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

- ***h(n)***: الكلفة التقديرية للوصول من العقدة n (قيد الدراسة)إلى الهدف
- ***g(n)***: كلفة الوصول إلى العقدة n.
- ***f(n)***: الكلفة الكلية التقديرية للممر عبر العقدة n وصولا إلى الهدف
- ***h(n)=0*** العقدة قيد البحث هي عقدة الهدف

مقارنة

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



- مقارنة بين البحث باستخدام خوارزميات الاستدلال و البحث باستخدام خوارزميات البحث الأعمى

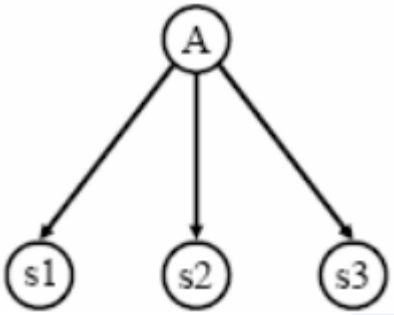
Heuristic search	Blind search
- يقوم بتقدير المسافة إلى الحالة الهدف - يرشد عمليات البحث باتجاه الهدف. - يميز الحالات الأقرب مسافة إلى الحالة الهدف.	- لديها معرفة فقط عن الحالات التي قامت باكتشافها من قبل. - لا توجد معرفة حول بعد أي حالة عن الحالة الهدف

الاجتهاد أو الاستكشاف أو الاستدلال
Heuristic

- **الاستدلال Heuristic:** هو تابع عندما يطبق على حالة يعيد رقم يقدر قرب هذه الحالة من الهدف.
- أي ان الاستدلال يخبر تقريبا كم تبقى للوصول الى الهدف (الأرقام الأصغر هي الأفضل)

أما بالنسبة للبحث في البيانات فإن الاستدلال يعني قواعد اختيار الفروع في فضاء الحالة للوصول إلى حل معقول للمسألة.

- مثال:
- للعقدة A ثلاث عناصر تالية بفرض التابع الحدسي لكل منها :
- $h(s1) = 0.8 \quad h(s2) = 2.0 \quad h(s3) = 1.6$
- متابعة البحث مرورا بالحالة s1 حدسيا هو الأفضل



Heuristic

- سوف نرى أن التجريبيات التي نعتبرها مقبولة admissible هي التجريبيات التي يكون تقديرها دائما أصغر من الكلفة الحقيقية .
- يستخدم الاجتهاد في حل مسائل الذكاء الصنعي في حالتين:
- **الأولى :** في حل المسائل التي ليس لها حل دقيق بسب الغموض المتأصل inherent ambiguities في المسألة أو البيانات المتوفرة. و نقصد بالغموض هنا إمكانية السير بمسارين مختلفين بنفس الشروط مما قد يقود إلى الابتعاد عن الحل.
- و يعتبر التشخيص الطبي أحد الأمثلة على ذلك, إذ يمكن أن يكون لنفس المجموعة من الأعراض أسباب مختلفة مما يدفع بالطبيب لاستخدام الاجتهاد لاختيار التشخيص الأرجح و يضع الخطة الملائمة للعلاج.

Heuristic

- و تعتبر الرؤية vision مثلا آخر على الحالات التي يظهر بها الالتباس أو الغموض و الذي يدفعنا إلى استخدام الاجتهاد, إذ تتصف المشاهد المرئية بالغموض و الالتباس مما يسمح بالتفسيرات المتعددة لصفات و حدود الكينونات المشاهدة, و يعتبر الخداع البصري في مقدمة هذه الالتباسات الغامضة. و هنا يأتي دور الاجتهاد ليجلو غموض المشهد و يرجح التفسير الذي يبدو أكثر منطقية.
- **الثانية:** نستخدم الاجتهاد عندما يكون هنالك نموذج رياضي للمسألة إلا أن العمل بهذا النموذج يجعلنا نعاني من كلفة (حسابية) فائقة قد تؤدي إلى جعل العمل به شبه مستحيل (كما هو عليه في مسائل الشطرنج التي توصلنا إلى الانفجار التراكبي (Combinatorial Explosion)) مما يجعل العمل بالاجتهاد أمرا لابد منه.
- **البحث الحدسي =البحث الاجتهادي = البحث الاستدلالي = البحث الاستكشافي = البحث التجريبي = Heuristic Search**

البحث بأفضل أول البسيط *greedy local search*
تسلق الهضبة (Hill climbing)

- يعتبر تسلق الهضبة أحد أبسط استراتيجيات البحث الاجتهادي. تعتمد هذه الاستراتيجية على توليد أبناء العقد و تقويم أبنائها, و انتقاء الابن الأفضل لتكرار عملية التوليد دون أخذ بقية العقد سواء كانت عقدة أب أو ابن أو عقدة أخوة أو غير ذلك بعين الاعتبار. و من الممكن الوصول إلى منطقة لا يمكن للمتسلق المتابعة بعدها وتدعى هذه النقطة قمة محلية Local Maxima و يتطلب من المتسلق في هذه الحالة الانتقال إلى طريق ممكن آخر, و يعتبر الوصول (الوقوع) في القمة المحلية أحد عيوب هذه الطريقة و بزيادة الخبرة يمكن تجنب الوصول إلى قمة محلية.

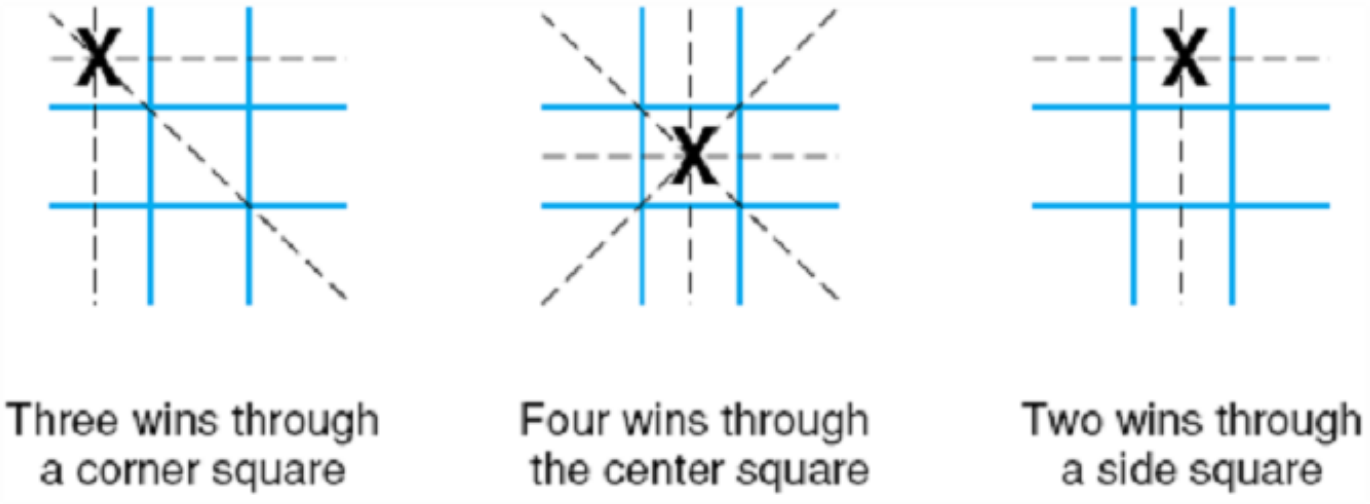
البحث بأفضل أول البسيط *greedy local search*
تسلق الهضبة (Hill climbing)

- إذ أن الخوارزمية الأولية لتسلق الهضبة لا تستطيع تذكر المراحل السابقة مما قد يؤدي إلى الحيرة و حتى إلى التوقف القسري عند الوصول إلى نقطة لا تسمح بالمتابعة لأنها تشكل تابع التقويم الأفضل, أي ليس هناك عقد ذات تابع تقويم أفضل.
- تعتبر لعبة Tic- Tac- Toe مثلا جيدا عن تسلق القمة (الهضبة), إذ يتبع اللاعب بشكل أولي الطريق الذي يقربه من الربح دون التروي و النظر في الحالة التي سيحققها المنافس, إذ ليس المطلوب في هذه الحالة تقصي الأخطاء و الاستفادة منها.

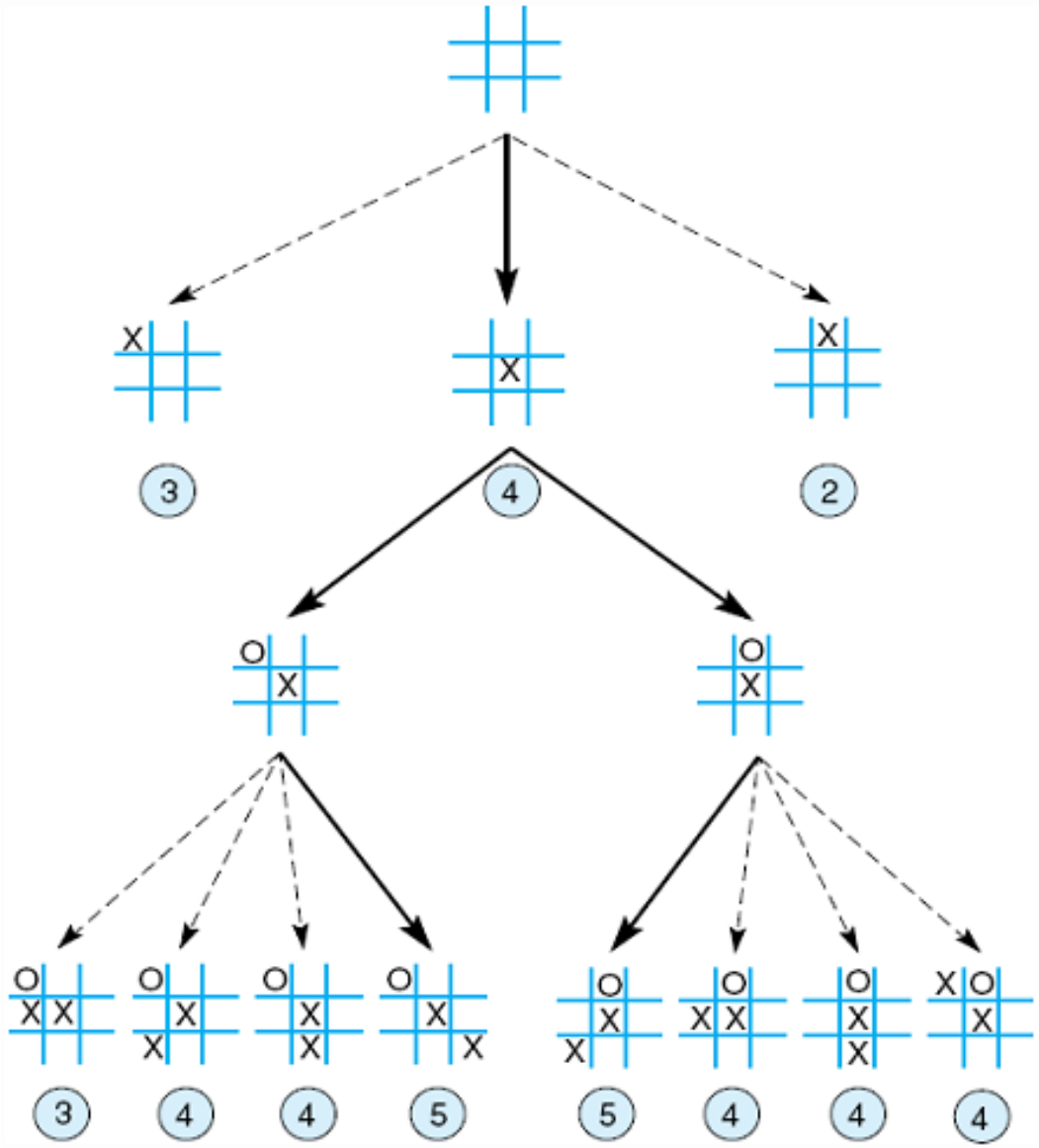
البحث بأفضل أول البسيط *greedy local search*
تسلق الهضبة (Hill climbing)

- لعبة Tic- Tac- Toe تشكل مسألة بفضاء حالة يصل إلى $9! = 362880$ حالة.
- و أحد طرق حلها يعتمد على فكرة القيام بالنقلة التي تقربنا من ربح محتمل.

- تابع التقويم المتبع هو حساب عدد خطوط الربح و اختيار النقلة ذات عدد خطوط الربح الأكبر كما هو مبين في الشكل.

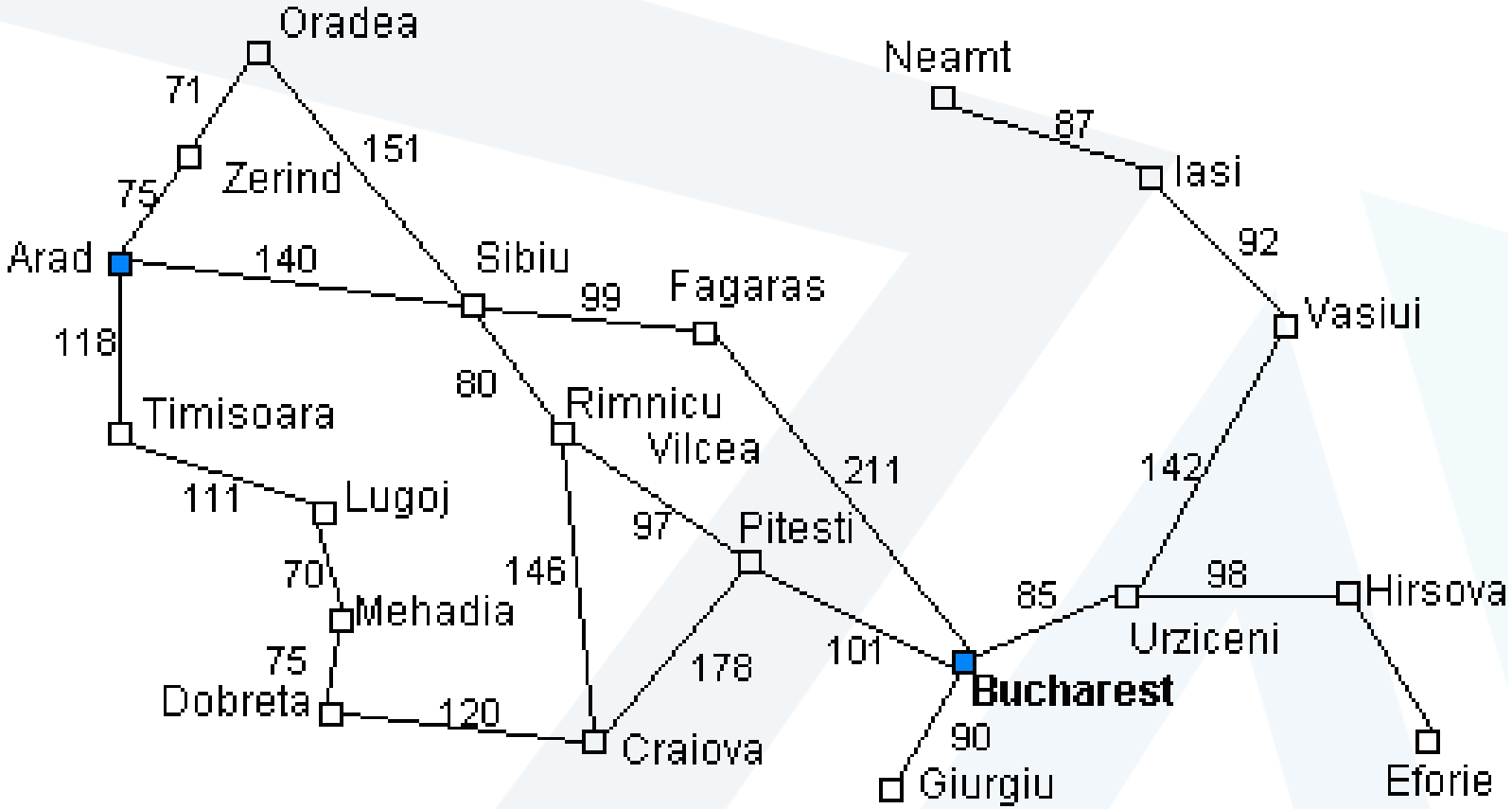


البحث بأفضل أول البسيط *greedy local search*
تسلق الهضبة (Hill climbing)



عندها يمكننا تقليل
عدد خطوات البحث
وفقا لهذا الاجتهاد

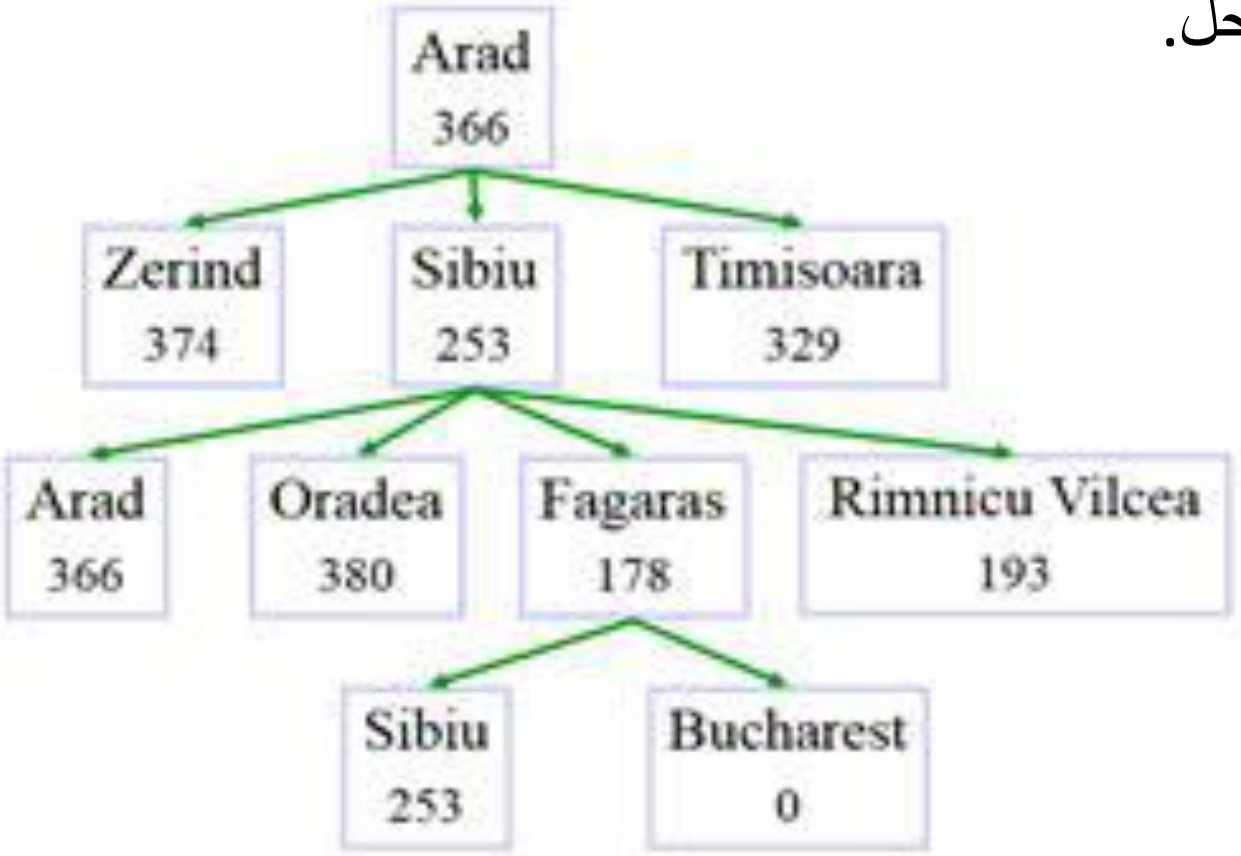
البحث بأفضل أول البسيط *greedy local search*
تسلق الهضبة (Hill climbing)



Straight-line distance to Bucharest	
Arad	366
Bucharest	0
Craiova	160
Dobreta	242
Eforie	161
Fagaras	178
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadia	241
Neamt	234
Oradea	380
Pitesti	98
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

البحث بأفضل أول البسيط *greedy local search*
تسلق الهضبة (Hill climbing)

هذه الشجرة هي عبارة عن شجرة الحل لمسألة خريطة رومانيا .يعبر الرقم الموجود في أسفل كل عقدة(مربع)عن الكلفة التقديرية للتجريبية(خط النظر .)ونلاحظ كيفية اختيار العقدة ذات الكلفة الأقل عند كل مستوى لنولد لها أبناء إلى أن نصل للحل.



البحث بأفضل أول البسيط *greedy local search*
تسلق الهضبة (Hill climbing)

- مثال : استخدام خوارزمية تسلق الهضبة (القمة) مع لعبة Pazzle -8:
- بدايةً دعونا نختار التجريبية الملائمة لطبيعة المسألة:
- 1-التجريبية الأولى:
- هي عدد المربعات التي لا تقع بمكانها الصحيح ففي المربع التالي ستكون قيمة التجريبية هي: H(n)=1 .

البحث بأفضل أول البسيط *greedy local search*
تسلق الهضبة (Hill climbing)

1	2	3
4	5	6
7	8	

Goal

1	2	3
4	5	6
7		8

- 2-التجريبية الثانية (Manhattan Distance) هي التي سنقوم باختيارها:

1	2	3
4	5	6
7		8

هي عدد الحركات اللازمة

لإيصال كل في غير موضعه

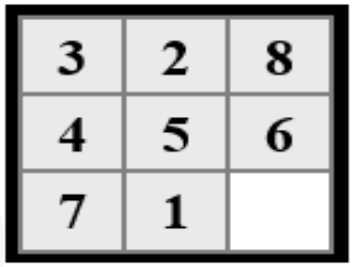
إلى مكانه الصحيح وستكون

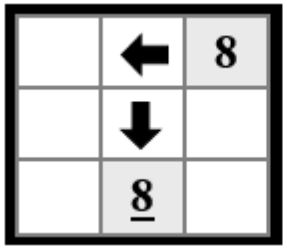
قيمة التجريبية هي H(n)=1 للمربع التالي

البحث بأفضل أول البسيط *greedy local search*
تسلق الهضبة (Hill climbing)

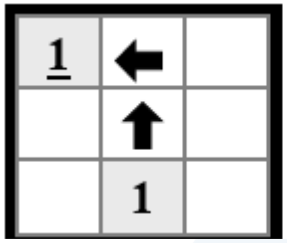
نلاحظ أنه كلاً من المربعات 1 , 3 , 8 ليست في مكانها الصحيح . وسنقوم بحساب عدد الحركات اللازمة لعودة كلاً من هذه المربعات لمكانها كما يلي:

Goal

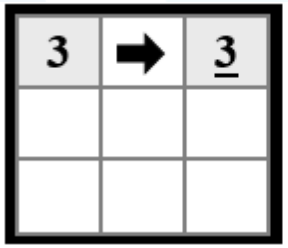




3 spaces



3 spaces

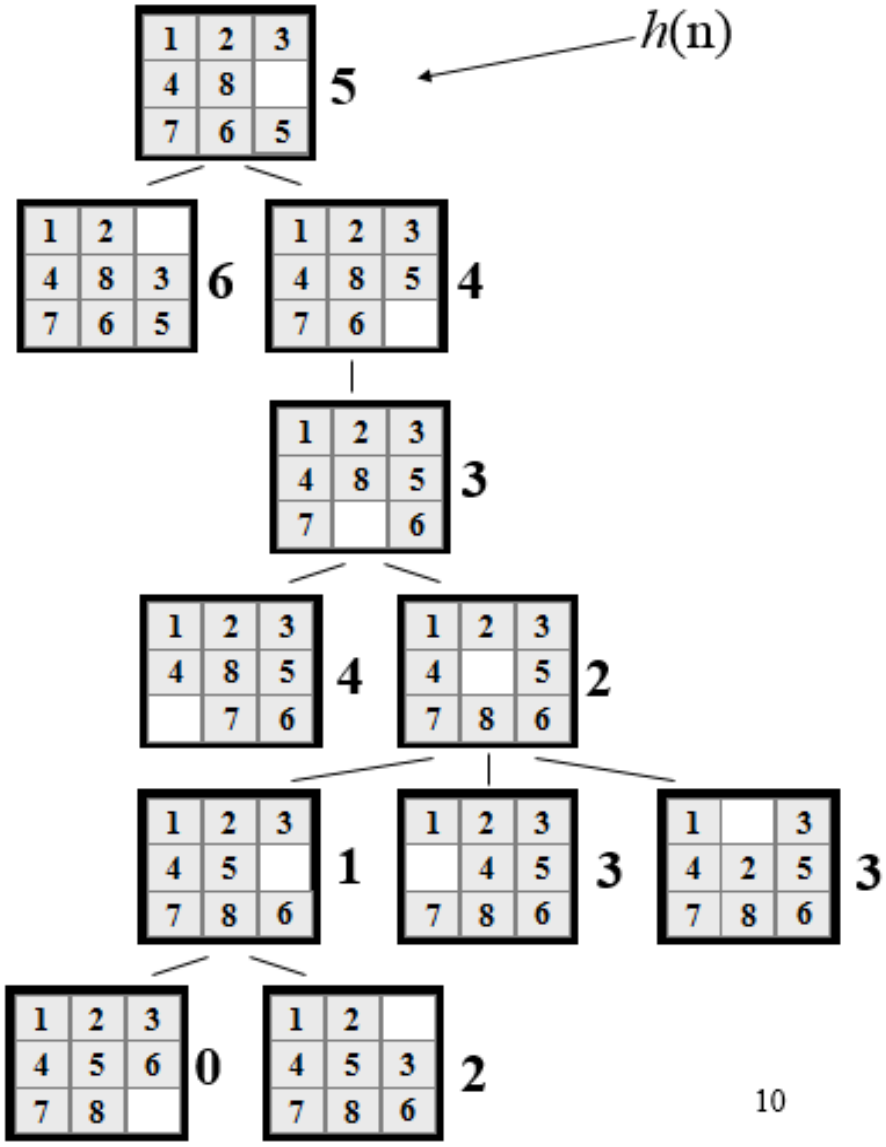


2 spaces

Total= 8 (للأشكال الثلاثة السابقة)

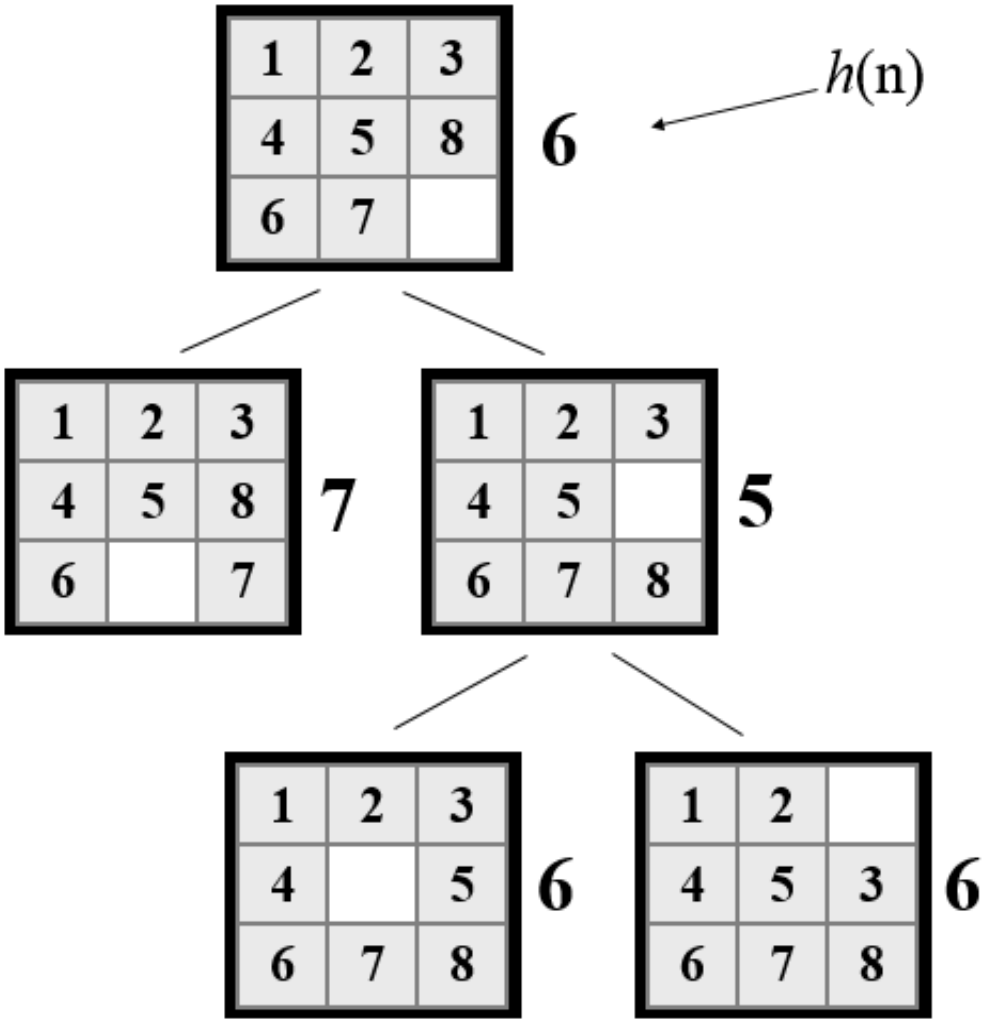
ونستطيع القول هنا أن التجريبية تفكر أنه بإمكاننا أن نصل للهدف من الخلال القيام ب 8 حركات فقط.

الحل



- يعبر الرقم بجانب كل رقعة عن قيمة تجريبية باستخدام Manhattan Distance .
- ولكن هناك مشكلة قد تواجهنا هي أن خوارزمية Hill Climbing لا تعطي دوماً حلاً أمثلياً كما في المثال التالي:

الحل



- نلاحظ في هذا المثال أن قيمة التجريبية ازدادت من مستوى لأخر بدلاً من أن تتناقص وبالتالي وقعت الخوارزمية في حالة Local Minima

خوارزمية عامة للبحث في البيان
GRAPH SEARCH

1- أنشئ شجرة بحث T_r مكونة فقط من عقدة البداية n_o .

ضع n_o في قائمة مرتبة تسمى OPEN .

2- أنشئ قائمة تسمى CLOSED تكون فارغة في البداية .

3- إذا كانت OPEN فارغة, اخرُج بالنتيجة: إخفاق.

4- اختر العقدة الأولى في OPEN, و احذفها من OPEN , و ضعها في CLOSED . سم هذه العقدة n .

5- إذا كانت n عقدة هدف, اخرج بنجاح مع الحل الذي حصلت عليه برسم مسار راجع على طول الأقواس في T_r من n إلى n_o (يتم انشاء الأقواس في الخطوة 6).

6- وسع العقدة n بتوليد مجموعة M من الخلَف. أرسِ M باعتبارها خلفا للعقدة n في T_r بإنشاء الأقواس من n إلى كل عنصر من M . ضع هذه العناصر من M في OPEN.

7- أعد ترتيب القائمة OPEN إما وفق بعض المخططات الاعتباطية أو وفق استحقاق تجريبي.

8- عد إلى الخطوة 3 .

خوارزمية عامة للبحث في البيان
GRAPH SEARCH

يمكن استعمال الخوارزمية السابقة لتنفيذ بحث الأفضل أولا , أو البحث عرضا أولا, أو البحث عمقا أولا. في البحث عرضا أولا نضع العقدة الجديدة في نهاية OPEN (الداخل أولا يخرج أولا أو FIFO) و لا يعاد ترتيب العقد. في البحث بطريقة العمق أولا نضع العقدة الجديدة في بداية OPEN (الداخل أخيرا يخرج أولا أو LIFO). في بحث الأفضل أولا (يسمى أيضا البحث التجريبي), يعاد ترتيب OPEN وفق استحقاق تجريبي للعقد.

البحث بأفضل أول
Search Best-first

و سنقدم فيما يلي البحث بأفضل أول بما يمكننا من تجاوز مسألة القمة المحلية Local Maxima

البحث بأفضل أول Best-first Search

نستخدم في هذه الحالة قائمتي OPEN و CLOSED كما هو عليه الحال في طرق البحث العشوائي. نضيف إلى فكرة اختيار الحل الأمثل تذكر الحالات السابقة التي تمكننا من الانتقال إلى الحالة الأفضل المتوفرة بين الحالات التي دخلت مرحلة OPEN كما يظهر من الخوارزمية التالية:

البحث بأفضل أول
Search Best-first

```
function best_first_search;

begin
   open := [Start];                                              % initialize
   closed := [ ];
   while open ≠ [ ] do                                           % states remain
      begin
         remove the leftmost state from open, call it X;
         if X = goal then return the path from Start to X
         else begin
                  generate children of X;
                  for each child of X do
                  case
                        the child is not on open or closed:
                           begin
                                 assign the child a heuristic value;
                                 add the child to open
                           end;
                        the child is already on open:
                           if the child was reached by a shorter path
                           then give the state on open the shorter path
                        the child is already on closed:
                           if the child was reached by a shorter path then
                              begin
                                 remove the state from closed;
                                 add the child to open
                              end;
                  end;                                           % case
                  put X on closed;
                  re-order states on open by heuristic merit (best leftmost)
               end;
return FAIL                                                      % open is empty
end.
```

(Greedy)best-first search
البحث(الجشع) بأفضل أول

- **تعتمد استراتيجيات البحث المعرفي على أسلوب البحث الجشع**
- **يمكننا أن نعرف البحث الجشع Greedy search أنه البحث الذي يحقق العلاقة $f(n) = h(n)$.**

تعريف تابع حدسي يخمن كل مسلك واعد h(n)

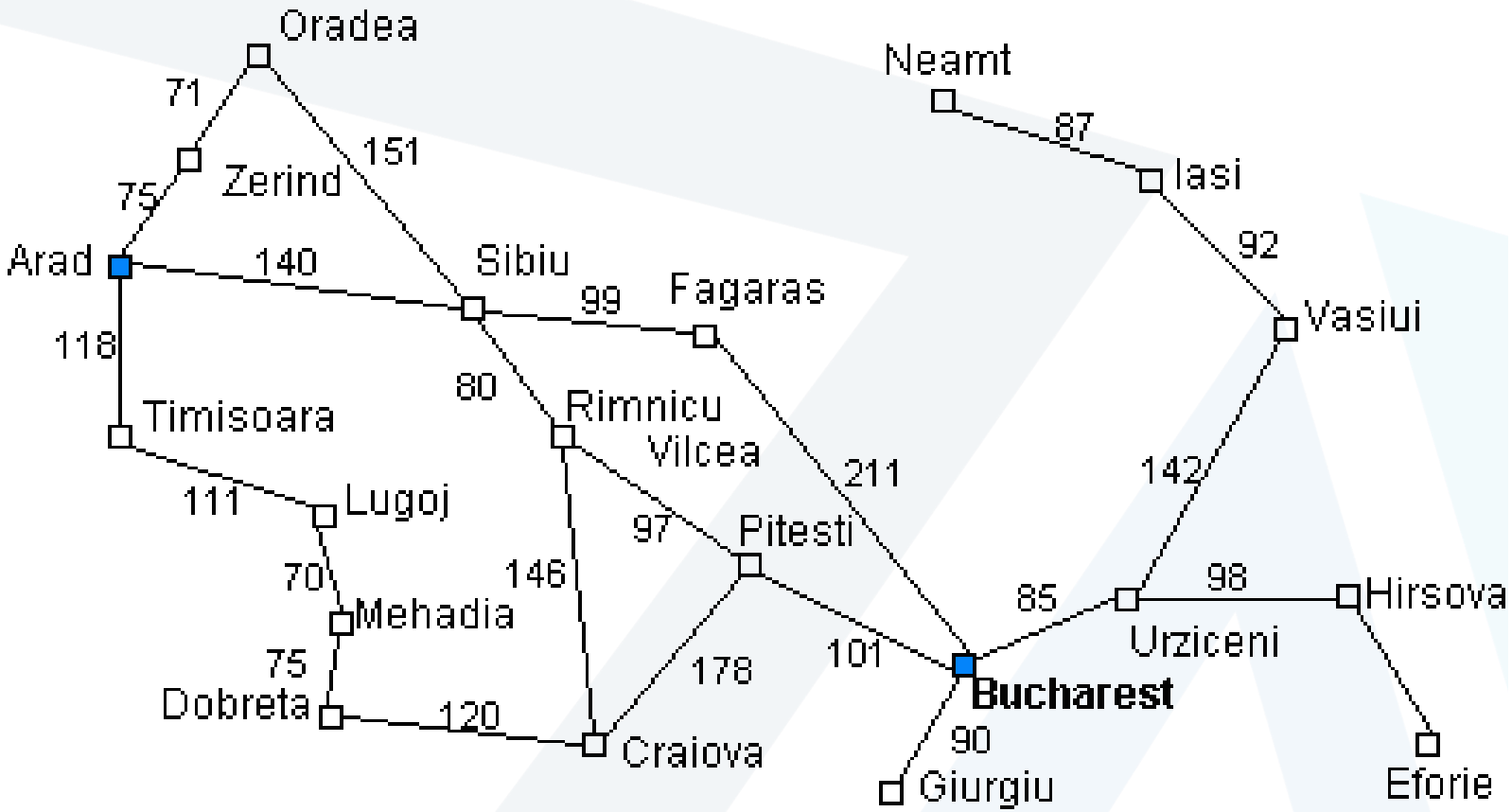
انطلاقا من عقدة ابتدائية سبر كل المسالك لاختيار المسلك الواعد

العقدة التي يبدو أنها الأكثر قربا من الهدف يكون لها أفضلية الاستكشاف

Greedy best-first search expands the node that appears to be closest to goal
Minimize the cost to reach a goal

في مثال السائح في رومانيا و بفرض البعد هو مسافة خط مستقيم (خط النظر) بين المدن هو الكلفة المتوقعة

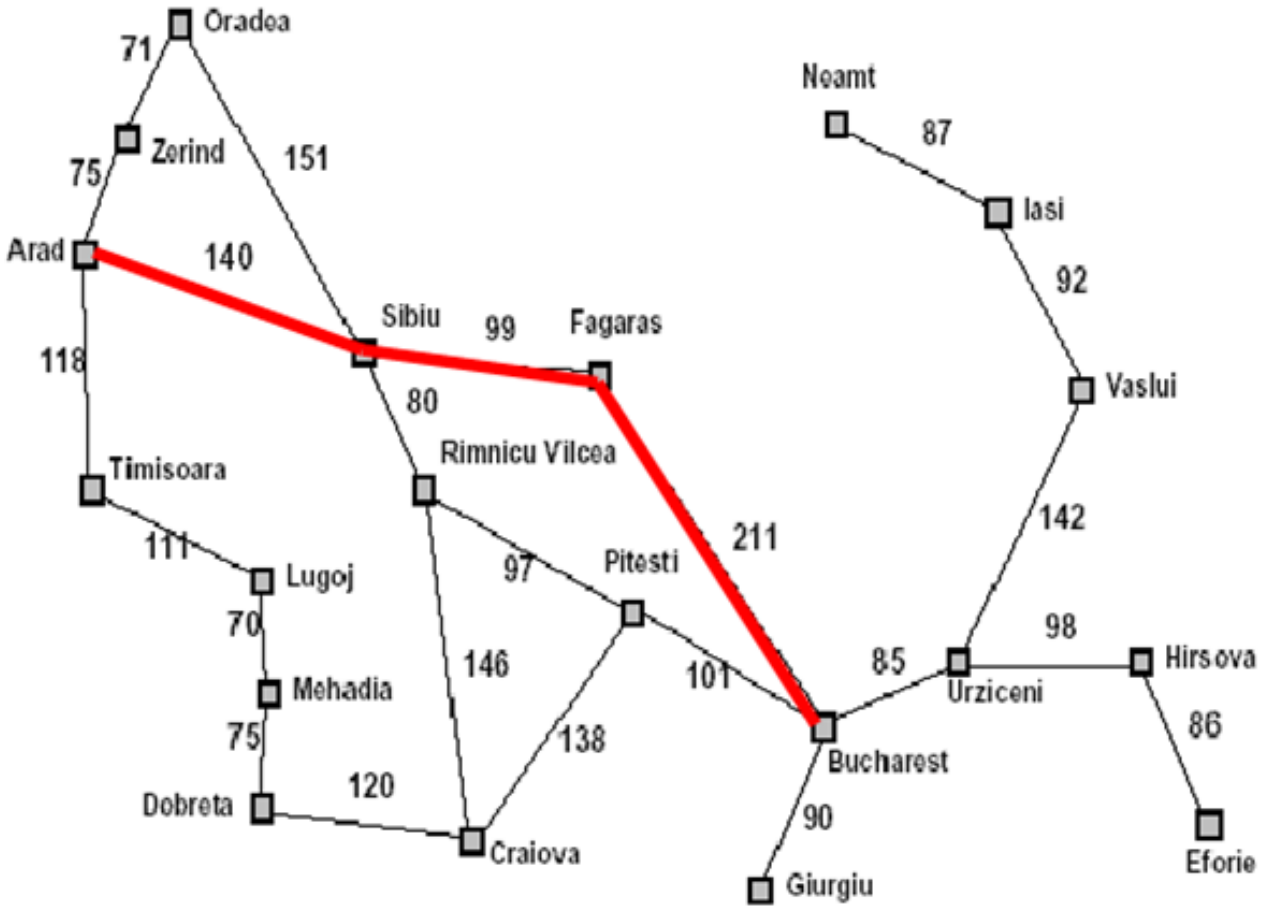
(Greedy)best-first search
البحث(الجشع) بأفضل أول



Straight-line distance to Bucharest	
Arad	366
Bucharest	0
Craiova	160
Dobreta	242
Eforie	161
Fagaras	178
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadia	241
Neamt	234
Oradea	380
Pitesti	98
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

Greedy)best-first search(

Example: traveler problem with straight-line distance information

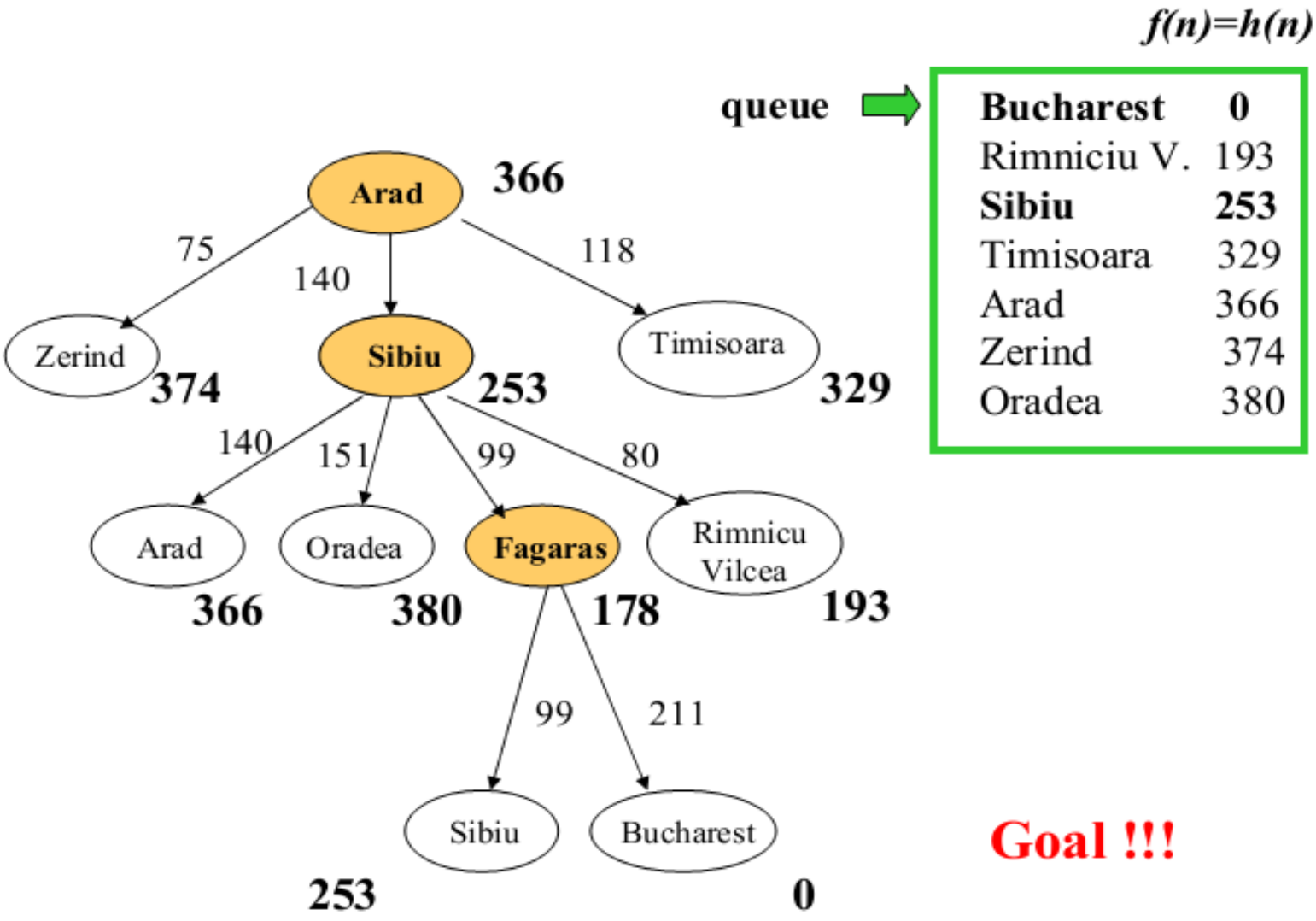


Straight-line distance to Bucharest	
Arad	366
Bucharest	0
Craiova	160
Dobreta	242
Eforie	161
Fagaras	178
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadia	241
Neamt	234
Oradea	380
Pitesti	98
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

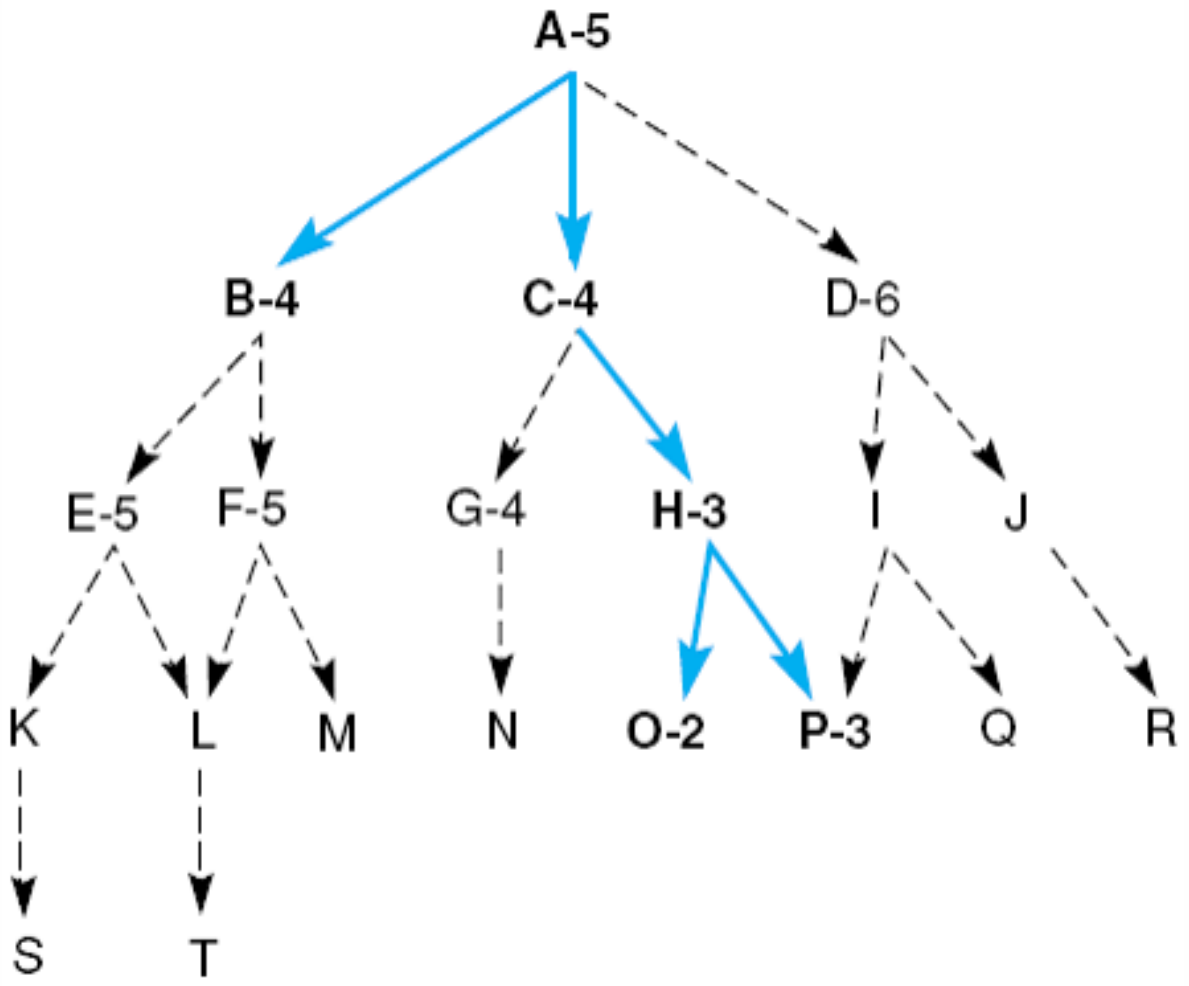
- **Greedy search result**

Greedy)best-first search(
البحث(الجشع) بأفضل أول

Greedy search



البحث بأفضل أولSearch Best-first



البحث بأفضل أولSearch Best-first

نتائج تقصي تطبيق الخوارزمية

1. **open = [A5]; closed = [ ]**
2. **evaluate A5; open = [B4,C4,D6]; closed = [A5]**
3. **evaluate B4; open = [C4,E5,F5,D6]; closed = [B4,A5]**
4. **evaluate C4; open = [H3,G4,E5,F5,D6]; closed = [C4,B4,A5]**
5. **evaluate H3; open = [O2,P3,G4,E5,F5,D6]; closed = [H3,C4,B4,A5]**
6. **evaluate O2; open = [P3,G4,E5,F5,D6]; closed = [O2,H3,C4,B4,A5]**
7. **evaluate P3; the solution is found!**

البحث بأفضل أولSearch Best-first