PROBLEMS, PROBLEM SPACES AND SEARCH

To Build a System to Solve a Particular Problem, The Following Four Things are Needed

1. Define the problem precisely- specify both initial and final situations(state)
2. Analyze the problem
3. Isolate and represent the task knowledge that is necessary to solve the problem
4. Choose the best problem solving technique and apply it

لبناء نظام لحل مشكلة معينة، هناك أربعة أمور مطلوبة:حدد المشكلة بدقة - حدد كل من المواقف الأولية والنهائية (الحالة)

حلل المشكلة

قم بعزل وتمثيل المعرفة المتعلقة بالمهمة اللازمة لحل المشكلة

اختر أفضل تقنية لحل المشكلة وقم بتطبيقها

State space search
Problem = Searching for a goal state

It is a process in which successive configurations or states of an instance are considered , with the goal of finding a goal state with a desired property

. State space- a set of states that a problem can be in.

- The group consisting of all the attainable states of a problem

ex: Customers in a line would have state space {0,1,2....}

البحث في مساحة الحالة

المشكلة = البحث عن حالة هدف إنها عملية يتم فيها النظر في التشكيلات أو الحالات المتعاقبة لمثيل ما، للعثور على حالة هدف ذات خاصية مرغوبة.مساحة الحالة هي مجموعة من الحالات حيث يمكن حل المشكلة. تتكون المجموعة من جميع الحالات التي يمكن الوصول إليها لمشكلة ما.على سبيل المثال: سيكون لدى العملاء في خط مساحة حالة {0، 1، 2، ....}

Search Algorithms

To successfully design and implement search algorithms, a programmer must be able to analyze and predict their behavior.

Many questions needed to be answered by the algorithm these include:

- Is the problem solver guaranteed to find a solution?
- Will the problem solver always terminate, or can it become caught in an infinite loop?
- When a solution is found, is it guaranteed to be optimal?
- What is the complexity of the search process in terms of time usage? Space search?
- How can the interpreter be designed to most effectively utilize a representation language?

- خوارزميات البحث
- لتصميم وتنفيذ خوارزميات البحث بنجاح، يجب أن يكون المبرمج قادرًا على تحليل وتوقع سلوكها.
- هناك العديد من الأسئلة التي تحتاج الخوارزمية إلى إجابة عليها، بما في ذلك:
- هل من المضمون أن يجد مُحلِّل المشكلة حلاً؟
- هل سينتهي حل المشكلة دائمًا، أم أنه قد يقع في حلقة لا نهائية؟
- عندما يتم العثور على حل، هل من المضمون أن يكون الحل الأمثل؟
- ما مدى تعقيد عملية البحث من حيث استخدام الوقت؟ مساحة البحث؟
- كيف يمكن تصميم المترجم للاستفادة بشكل أكثر فعالية لتمثيل اللغة؟

State Space Search

- The theory of state space search is our primary tool for answering these questions, by representing a problem as state space graph, we can use graph theory to analyze the structure and complexity of both the problem and procedures used to solve it.

Graph Theory:-

- A graph consists of a set of a nodes and a set of arcs or links connecting pairs of nodes. The domain of state space search, the nodes are interpreted to be stated in problem solving process, and the arcs are taken to be transitions between states.

Graph theory is our best tool for reasoning about the structure of objects and relations.

البحث في فضاء الحالة

إن نظرية البحث في فضاء الحالة هي أداتنا الأساسية للإجابة على هذه الأسئلة، وذلك من خلال تمثيل المشكلة على هيئة رسم بياني لفضاء الحالة. ويمكننا استخدام نظرية الرسم البياني لتحليل بنية وتعقيد كل من المشكلة والإجراءات المستخدمة لحلها.

نظرية الرسم البياني: يتكون الرسم البياني من مجموعة من العقد ومجموعة من الأقواس أو الروابط التي تربط أزواج العقد. وفي مجال البحث في فضاء الحالة، يتم تفسير العقد على أنها مذكورة في عملية حل المشكلة، ويتم اعتبار الأقواس بمثابة انتقالات بين الحالات.

إن نظرية الرسم البياني هي أفضل أداة لدينا للتفكير في بنية الأشياء والعلاقات.

Representing search problems

Using directed graph

- The states are represented as nodes
- The allowed actions are represented as arcs.

تمثيل مشكلات البحث باستخدام الرسم البياني الموجه:
يتم تمثيل الحالات على هيئة عقد
يتم تمثيل الإجراءات المسموح بها على هيئة أقواس.

Problem Formulation

- A single state problem formulation is defined by four items

  Initial state, successor function, goal test and path cost

- Problem formulation means choosing a relevant set of states to consider, and a feasible set of operators for moving from one state to another

- Search is the process of imagining sequences of operators applied to the initial state and checking which sequence reaches a goal state.

صياغة المشكلة

يتم تعريف صياغة مشكلة الحالة الواحدة بأربعة عناصر:الحالة الأولية، ووظيفة الخلف، واختبار الهدف وتكلفة المسار

تعني صياغة المشكلة اختيار مجموعة ذات صلة من الحالات للنظر فيها، ومجموعة قابلة للتطبيق من المشغلات للانتقال من حالة إلى أخرى.

البحث هو عملية تخيل تسلسلات المشغلات المطبقة على الحالة الأولية والتحقق من التسلسل الذي يصل إلى حالة الهدف.

Examples.

Problem: On holiday in Singapur; currently in Mysur. Flight leaves tomorrow from Bangalore. Find a short route to drive to Bangalore.

Formulate problem:

- **states:** various cities
- **actions:** drive between cities
- **solution:** sequence of cities
- **Path Cost:** distance travelled

المشكلة:

أقضي إجازة في سنغافورة؛ وأقيم حاليًا في ميسورو. تغادر الرحلة غدًا من بنغالور. ابحث عن طريق قصير للقيادة إلى بنغالور.

صِغ المشكلة:

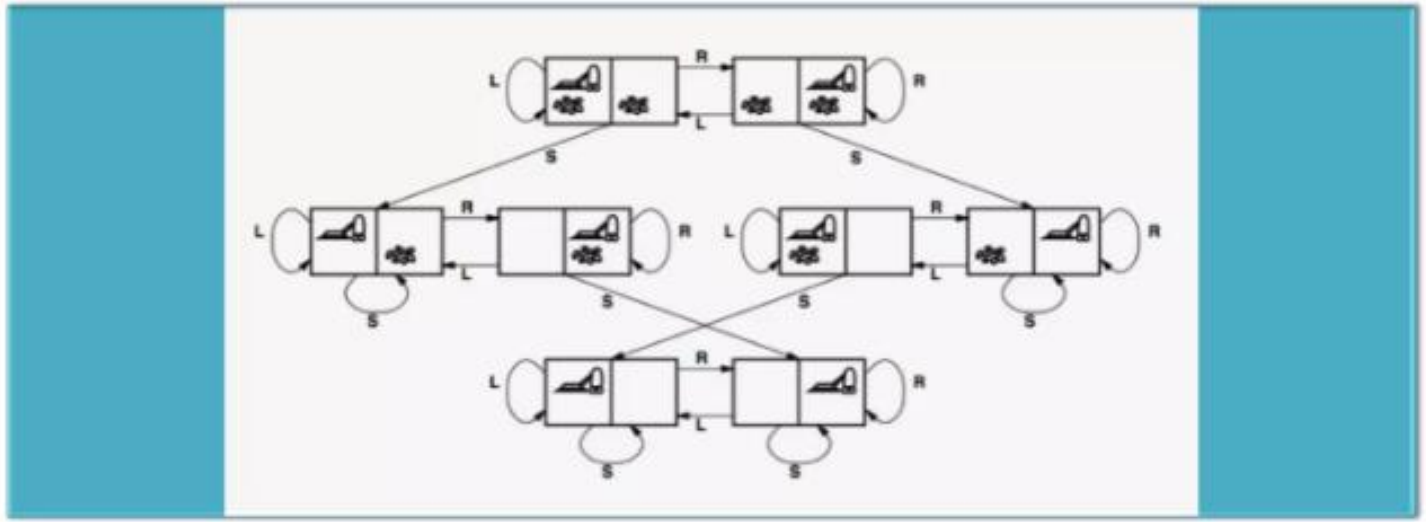
الولايات: مدن مختلفة

الإجراءات: القيادة بين المدن

الحل: سلسلة من المدن

مسار التكلفة: المسافة المقطوعة

Vacuum world state space

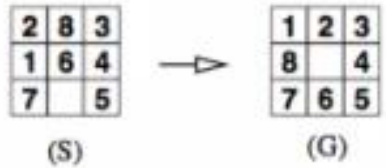


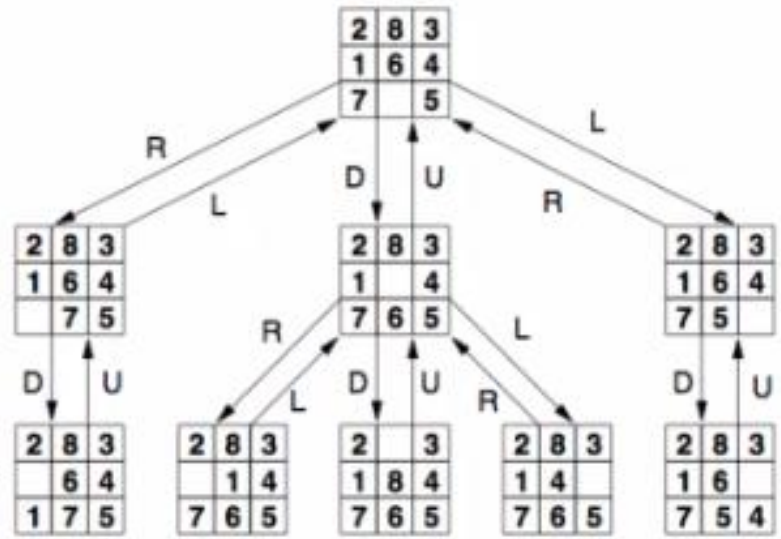
States: Dirt and Robot Location
Actions: Left, right, clean
Goal test: No dirt at all locations
Path cost: 1 per action

فراغ حالة العالم الفضاءالحالات:
الأوساخ وموقع الروبوت
الإجراءات: اليسار، اليمين، التنظيف
اختبار الهدف: لا توجد أوساخ في جميع المواقع
تكلفة المسار: 1 لكل إجراء

The 8 - Puzzle

Problem: Go from state S to state G.





States: Locations of tiles
Actions: Move blank left, right, up, down
Goal test: Given
Path cost: 1 per move

State space search: Playing Chess

- Each position can be described by an 8 by 8 array.
- Initial position is the game opening position.
- Goal position is any position in which the opponent does not have a legal move and his or her king is under attack.
- Legal moves can be described by a set of rules:

-Left sides can be described by a set of rules

-Right sides describe the new resulting state

لعب الشطرنج

يمكن وصف كل موضع بمصفوفة 8 × 8.

الموضع الأولي هو موضع افتتاح اللعبة.

موضع الهدف هو أي موضع لا يملك فيه الخصم حركة قانونية ويكون ملكه تحت الهجوم.

يمكن لمجموعة من القواعد وصف الحركات القانونية:

يمكن وصف الجوانب اليسرى بمجموعة من القواعد

تصف الجوانب اليمنى الحالة الناتجة الجديدة

Playing chess contd...

- State space is a set of legal positions.
- Starting at the initial state.
- Using the set of rules to move from one state to another.
- Attempting to end up in a goal state.

استمرار لعب الشطرنج...مساحة الحالة عبارة عن مجموعة من المواقف القانونية.البدء من الحالة الأولية.استخدام مجموعة القواعد للانتقال من حالة إلى أخرى.محاولة الوصول إلى حالة الهدف.

Playing Chess Contd..

One Legal Chess move

Playing Chess Contd..

- Writing the rules like above leads to very large number
- These rule poses serious practical difficulties
- No person could ever supply a complete set of rules. It would take too long and could certainly not be done without mistakes
- No program could easily handle all those rules.

متابعة لعب الشطرنج..إن كتابة القواعد كما هو موضح أعلاه تؤدي إلى عدد كبير جدًا.هذه القواعد تفرض صعوبات عملية خطيرة.لا يمكن لأي شخص على الإطلاق أن يقدم مجموعة كاملة من القواعد. سيستغرق الأمر وقتًا طويلاً ومن المؤكد أنه لا يمكن القيام بذلك دون أخطاء.لا يمكن لأي برنامج التعامل بسهولة مع كل هذه القواعد.

Playing Chess Contd..

Another way to describe the chess moves

White pawn at **Square(file e, rank 2)** **AND** **Square(file e, rank 3) is empty** **AND** **Square(file e, rank 4) is empty**	→	**Move a pawn from** **Square(file e, rank 2)** **to** **Square(file e, rank 4)**

- Irrecoverable problem: in which solution steps cannot be undone.

Ex:- Chess

Suppose a chess playing program makes a stupid move and realize it a couple of move later. It cannot simply play as though it never made the stupid move. Nor can it simply backup and start the game over from that point. All it can do is to try to make best of the current situation and go on from there.

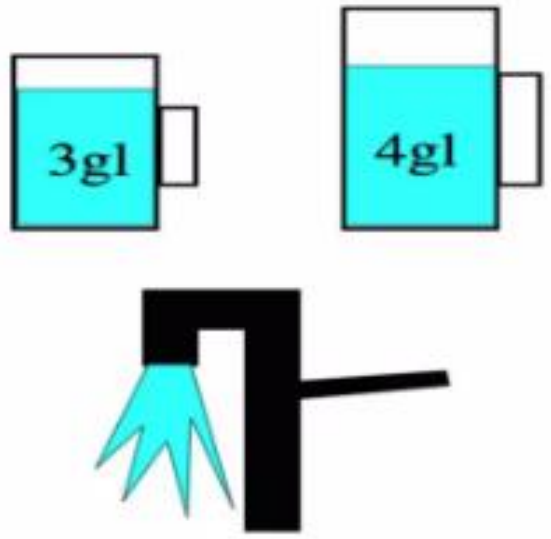
هل يمكن تجاهل خطوات الحل أو التراجع عنها؟مشكلة لا يمكن إصلاحها: حيث لا يمكن التراجع عن خطوات الحل.مثال: الشطرنج لنفترض أن برنامجًا للعب الشطرنج قام بحركة غبية وأدرك ذلك بعد حركتين. لا يمكنه ببساطة اللعب كما لو أنه لم يقم بالحركة الغبية أبدًا. ولا يمكنه ببساطة النسخ الاحتياطي وبدء اللعبة من تلك النقطة. كل ما يمكنه فعله هو محاولة الاستفادة القصوى من الموقف الحالي والمضي قدمًا من هناك.

The Water Jug Problem

Given two jugs, a 4-gallon and 3-gallon, neither has any measuring maskers on it. There is a pump that can be used to fill the jugs with water. How can you get exactly 2 gallons of water into the 4-gallon jug?
1 Gallon = 3.785 Liter

ما هي مشكلة جرة الماء في الذكاء الاصطناعي؟
مشكلة جرة الماء في الذكاء الاصطناعي هي لغز كلاسيكي في الذكاء الاصطناعي والرياضيات يركز على تحسين استخدام جرة ماء أو أكثر لقياس كمية معينة من الماء. إنها مشكلة أساسية في مجال التحسين واتخاذ القرار. تأتي هذه المشكلة بأشكال مختلفة مع سعات جرة مختلفة وقياسات مستهدفة، مما يجعلها أداة متعددة الاستخدامات لتعلم تقنيات حل المشكلات في الذكاء الاصطناعي.

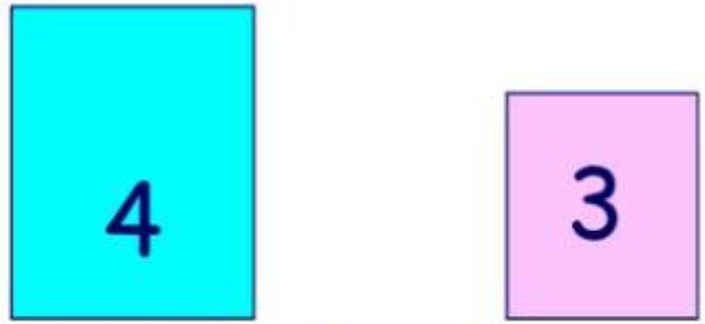
تتضمن مشكلة إبريق الماء في الذكاء الاصطناعي قيودًا وأهدافًا تجعلها لغزًا:القيد 1: تتمتع الأباريق بسعات محدودة.القيد 2: لا يمكنك ملء أو صب الماء إلا بين الأباريق أو من المصدر.الهدف: الهدف هو قياس كمية معينة من الماء بدقة، عادةً عن طريق الجمع بين الماء ونقله بين الأباريق.



Get exactly 2 gallons of water into the 4gl jug.

Defining the problem

- A water jug problem: 4-gallon and 3-gallon



- no marker on the bottle
- pump to fill the water into the jug
- How can you get exactly 2 gallons of water into the 4-gallons jug?

A state space search

(x,y) : order pair

x : water in 4-gallons → x = 0,1,2,3,4

y : water in 3-gallons → y = 0,1,2,3

start state : (0,0)

goal state : (2,n) where n = any value

Rules :

1. Fill the 4 gallon-jug (4,-)
2. Fill the 3 gallon-jug (-,3)
3. Empty the 4 gallon-jug (0,-)
4. Empty the 3 gallon-jug (-,0)

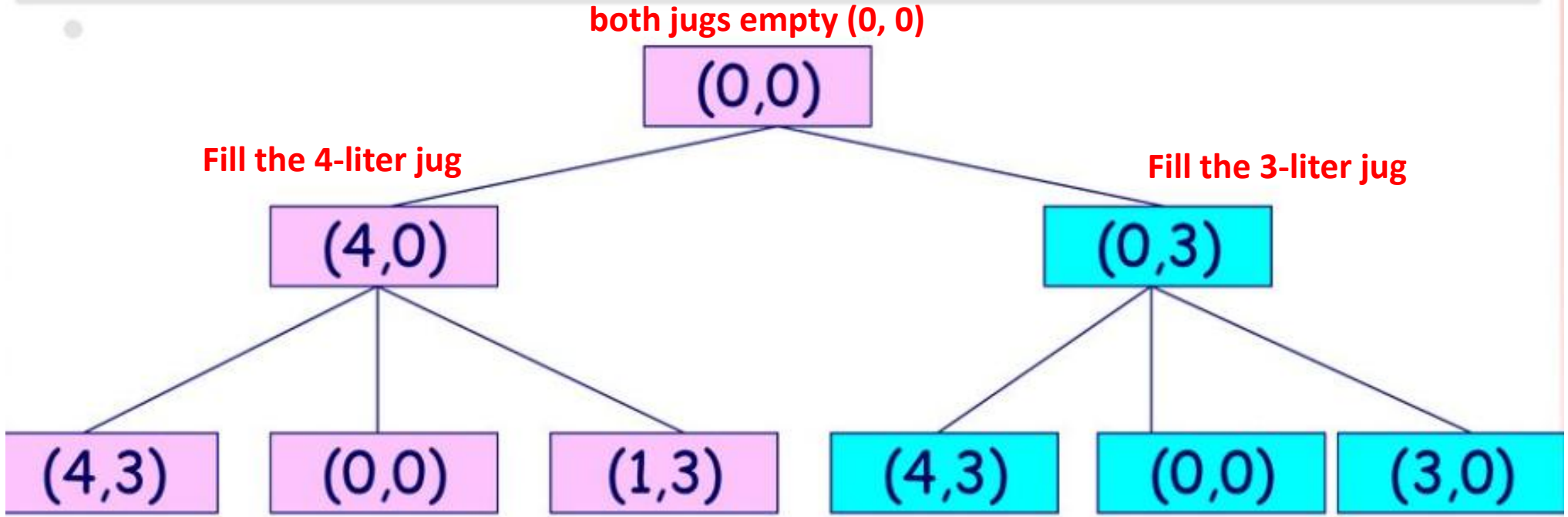
Water jug rules

1	(x, y) if $x < 4$	$\rightarrow (4, y)$	Fill the 4-gallon jug
2	(x, y) if $y < 3$	$\rightarrow (x, 3)$	Fill the 3-gallon jug
3	(x, y) if $x > 0$	$\rightarrow (x - d, y)$	Pour some water out of the 4-gallon jug
4	(x, y) if $y > 0$	$\rightarrow (x, y - d)$	Pour some water out of the 3-gallon jug
5	(x, y) if $x > 0$	$\rightarrow (0, y)$	Empty the 4-gallon jug on the ground
6	(x, y) if $y > 0$	$\rightarrow (x, 0)$	Empty the 3-gallon jug on the ground
7	(x, y) if $x + y \geq 4$ and $y > 0$	$\rightarrow (4, y - (4 - x))$	Pour water from the 3-gallon jug into the 4-gallon jug until the 4-gallon jug is full

Water jug rules

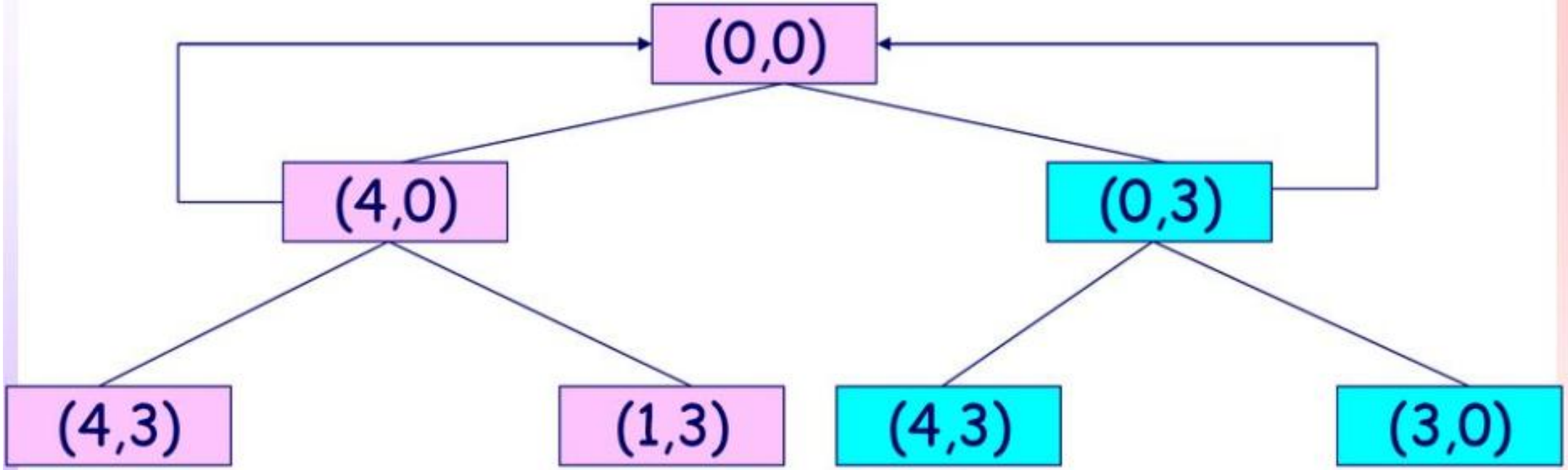
8	(x, y) if $x + y \geq 3$ and $x > 0$	$\rightarrow (x - (3 - y), 3)$	Pour water from the 4-gallon jug into the 3-gallon jug until the 3-gallon jug is full
9	(x, y) if $x + y \leq 4$ and $y > 0$	$\rightarrow (x + y, 0)$	Pour all the water from the 3-gallon jug into the 4-gallon jug
10	(x, y) if $x + y \leq 3$ and $x > 0$	$\rightarrow (0, x + y)$	Pour all the water from the 4-gallon jug into the 3-gallon jug
11	$(0, 2)$	$\rightarrow (2, 0)$	Pour the 2 gallons from the 3-gallon jug into the 4-gallon jug
12	$(2, y)$	$\rightarrow (0, y)$	Empty the 2 gallons in the 4-gallon jug on the ground

Search Tree



Water jug problem.

Search Graph



- Water jug problem.
  - Cycle
  - When will the search terminate?

The Water Jug Problem

The state space for this problem can be described as the set of ordered pairs of integers (x,y) such that x = 0, 1,2, 3 or 4 and

y = 0,1,2 or 3; x represents the number of gallons of water in the 4-gallon jug and y represents the quantity of water in 3-gallon jug

The start state is (0,0)

The goal state is (2,n)

Production rules for Water Jug Problem

The operators to be used to solve the problem can be described as follows:

Sl No	Current state	Next State	Descritpion
1	(x,y) if x < 4	(4,y)	Fill the 4 gallon jug
2	(x,y) if y <3	(x,3)	Fill the 3 gallon jug
3	(x,y) if x > 0	(x-d, y)	Pour some water out of the 4 gallon jug
4	(x,y) if y > 0	(x, y-d)	Pour some water out of the 3-gallon jug
5	(x,y) if x>0	(0, y)	Empty the 4 gallon jug
6	(x,y) if y >0	(x,0)	Empty the 3 gallon jug on the ground
7	(x,y) if x+y >= 4 and y >0	(4, y-(4-x))	Pour water from the 3 – gallon jug into the 4 –gallon jug until the 4-gallon jug is full

Production rules

8	(x, y) if x+y >= 3 and x>0	(x-(3-y), 3)	Pour water from the 4-gallon jug into the 3-gallon jug until the 3-gallon jug is full
9	(x, y) if x+y <=4 and y>0	(x+y, 0)	Pour all the water from the 3-gallon jug into the 4-gallon jug
10	(x, y) if x+y <= 3 and x>0	(0, x+y)	Pour all the water from the 4-gallon jug into the 3-gallon jug
11	(0,2)	(2,0)	Pour the 2 gallons from 3-gallon jug into the 4-gallon jug
12	(2,y)	(0,y)	Empty the 2 gallons in the 4-gallon jug on the ground

To solve the water jug problem

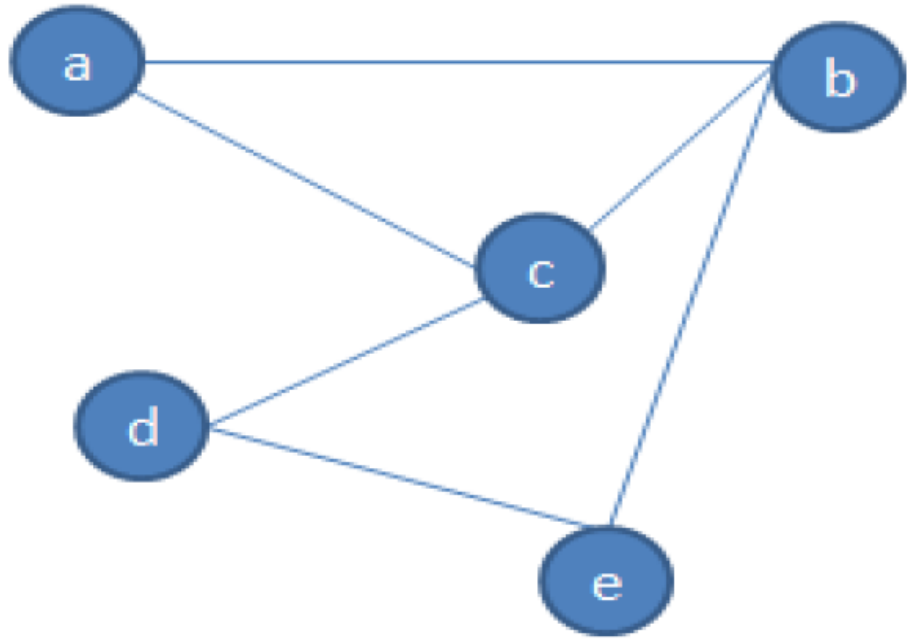
- Required a control structure that loops through a simple cycle in which some rule whose left side matches the current state is chosen
- the appropriate change to the state is made as described in the corresponding right side
- the resulting state is checked to see if it corresponds to goal state.

One solution to the water jug problem

Shortest such sequence will have a impact on the choice of appropriate mechanism to guide the search for solution.

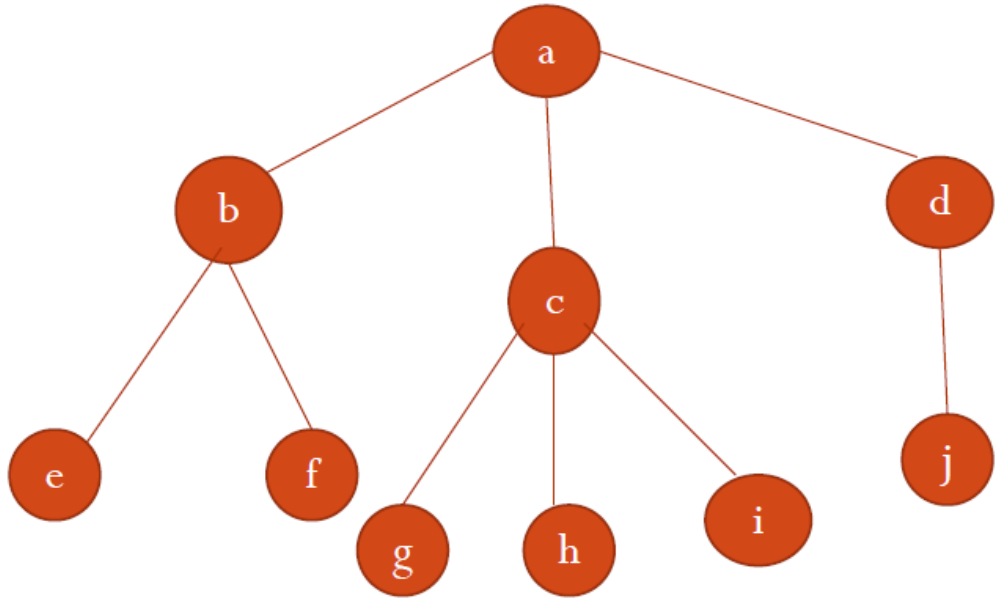
Gallons in the 4-gallon jug	Gallons in the 3-gallon jug	Rule applied
0	0	2
0	3	9
3	0	2
3	3	7
4	2	5 or 12
0	2	9 0r 11
2	0	

Graph Theory



Nodes={a,b,c,d,e}

Arcs={(a,b), (a,c),(b,c),(b,e),(d,e),(d,c),(e,d)}



Nodes={a,b,c,d,e,f,g,h,i,j}
Arcs={(a,b),(a,c),(a,d),(b,e),(b,f),(c,g),(c,h),(c,i),(d,j)}

State Space Representation

A state space is represented by four tuple [N,A,S,GD], where:-

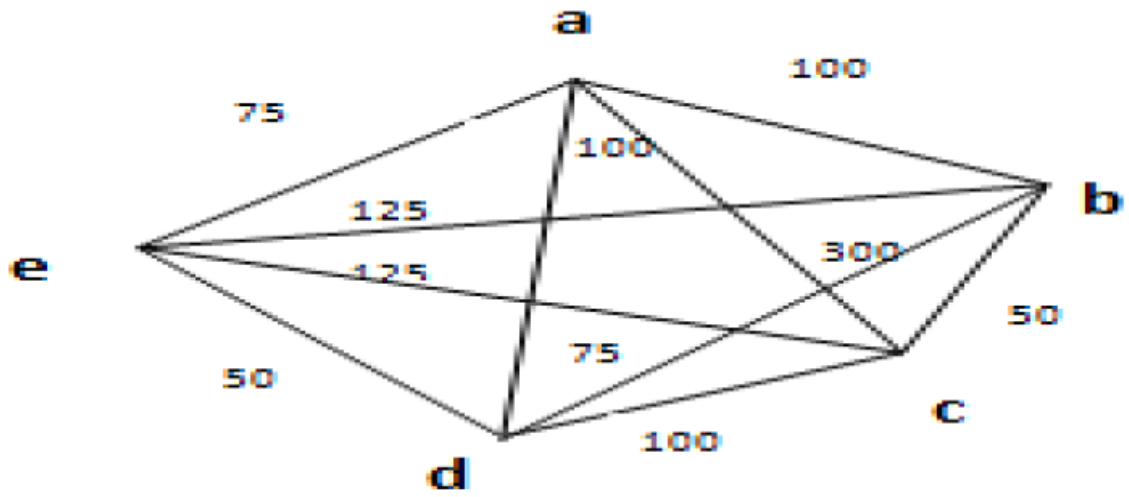
- **N** is a set of nodes or states of the graph. These correspond to the states in a problem –solving process.
- **A** is the set of arcs between the nodes. These correspond to the steps in a problem –solving process.
- **S** a nonempty subset of N , contains the start state of the problem.

- **G** a nonempty subset of N contains the goal state of the problem.

- **A solution path:- Is a path through this graph from a node S to a node in GD.**

Example:- Traveling Salesman Problem

- Starting at A , find the shortest path through all the cities , visiting each city exactly once returning to A.



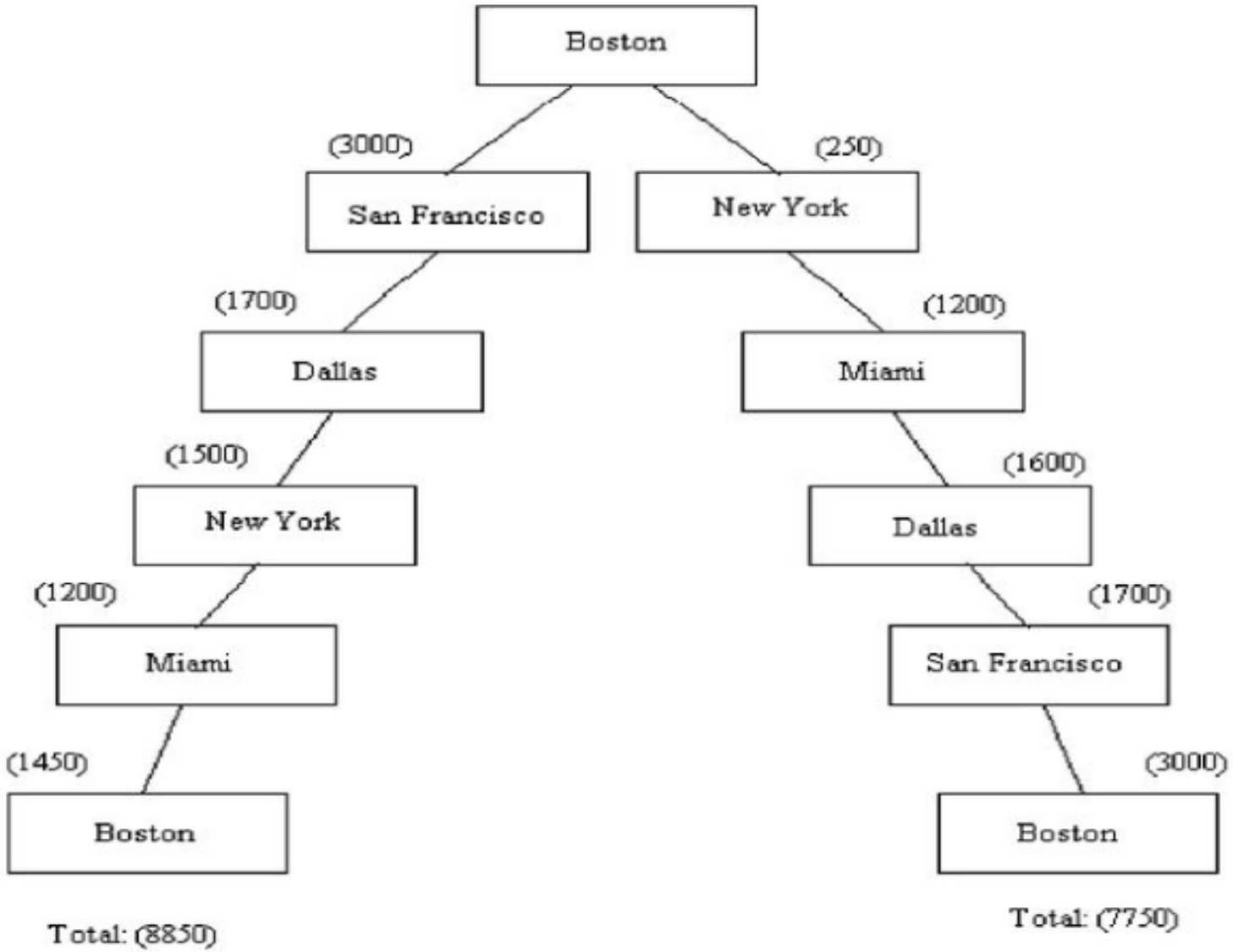
The complexity of exhaustive search in the traveling Salesman is **(N-1)!**, where N is the No. of cities in the graph. There are several technique that reduce the search complexity.

Best-path problem

Ex: Traveling Salesman Problem

- Given a road map of n cities, find the **shortest** tour which visits every city on the map exactly once and then return to the original city (*Hamiltonian circuit*)

	Boston	New York	Miami	Dallas	S.F.
Boston		250	1450	1700	3000
New York	250		1200	1500	2900
Miami	1450	1200		1600	3300
Dallas	1700	1500	1600		1700
S.F.	3000	2900	3300	1700	


Boston
(3000)
San Francisco
(250)
New York
(1700)
Dallas
(1200)
Miami
(1500)
New York
(1600)
Dallas
(1200)
Miami
(1700)
San Francisco
(1450)
Boston
(3000)
Boston
Total: (8850)
Total: (7750)

Any-path problem

Ex: Answer-question System

Consider the problem of answering the question based on following facts:

1. Marcus was a man.
2. Marcus was a Pompean.
3. Marcus was born in 40 AD.
4. All men are mortal.
5. All Pompeans died when volcano erupted in 79 AD.
6. No mortal lives longer than 150 years.
7. Now it is 1991 AD.

“ Is Marcus alive?”

مسألة أي مسارمثال:
نظام الإجابة على السؤال
فكر في مسألة الإجابة على السؤال بناءً على الحقائق التالية:

1. كان ماركوس رجلاً.
2. كان ماركوس من بومبي.
3. ولد ماركوس في عام 40 بعد الميلاد.
4. كل الرجال فانون.
5. مات كل بومبي عندما ثار البركان في عام 79 بعد الميلاد.
6. لا يعيش أي بشر أكثر من 150 عامًا.
7. الآن نحن في عام 1991 بعد الميلاد.
8. "هل ماركوس على قيد الحياة؟"

1. Marcus was a man.
2. Marcus was a Pompeian.
3. Marcus was born in 40 A.D.
4. All men are mortal.
5. All Pompeians died when the volcano erupted in 79 A.D.
6. No mortal lives longer than 150 years.
7. It is now 1991 A.D.

Is Marcus alive?

Solution 1

1. Marcus was man.	axiom 1
4. All men are mortal.	axiom 4
8. Marcus is mortal.	1,4
3. Marcus was born in 40 A.D.	axiom 3
7. It is now 1991 A.D.	axiom 7
9. Marcus' age is 1951 years.	3,7
6. No mortal lives longer than 150 years	axiom 6
10. Marcus is dead.	8,6,9

تقدم الصورة عملية استنتاج منطقية لاستنتاج ما إذا كان ماركوس حيًا أم ميتًا، استنادًا إلى سلسلة من البديهيات وخطوات الاستدلال. فيما يلي تفصيل مفصل:

Solution 2

7. It is now 1991 A.D.	axiom 7
5. All Pompeians died in 79 A.D.	axiom 5
11. All Pompeians are dead now.	7,5
2. Marcus was a Pompeian.	axiom 2
12. Marcus is dead.....	11,2