

Chapter 3

في الشاتر ده هنتكلم عن ال solving problem وخاصة اللي by search
وهنعرف أي ال Problems اللي محتاجة سيرش ولا مش محتاجة

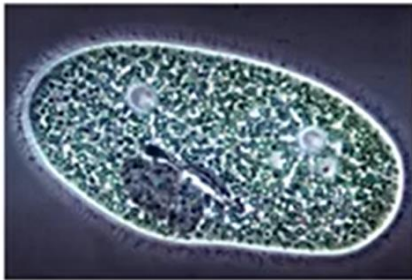
Solving Problems by SEARCHING



نبدأ من اننا نعرف أي أنواع ال Agents اللي عندنا

Types of Agents

α Reflex Agent :



Considers how the world IS

- Choose action based on current percept.
- Do not consider the future consequences of actions.

α Planning agent :



Considers how the world WOULD BE

- Decisions based on (hypothesized) consequences of actions.
- Must have a model of how the world evolves in response to actions.
- Must formulate a goal.

- عندنا ال reflexAgent ده بيقى مبني على قرار هتخدو دلوقتي ملهوش دعوة بالماضي ولا المستقبل ومش شرط بيبقى متخططلوزي انت وصحبك في القهوة كده وبيقول يلا ساحل لا عارف أي اللي هيحصل ولا هو معاه فلوس ولا أي اللي ممكن يجرا لو روحنا دلوقتي وهكذا قولت ادي مثال بماننا في سمر وكده واه طبيعي ان نوع ال reflex agent ده مبنستخداموش غي حياتنا او علشان نحل مشكلة ولا فيه سيرش ولا حاجة
- و الثاني ال planning agent ده بقى اللي بيبقى بتخطيط ونبص على اللي ورا وقد يتبني عليه المستقبل وبنسخدمو في حل المشاكل ب ال algorithms والسيرش

State Space Search

Problems are solved by searching among alternative choices.

○ Humans consider several alternative strategies on their way to solving a problem.

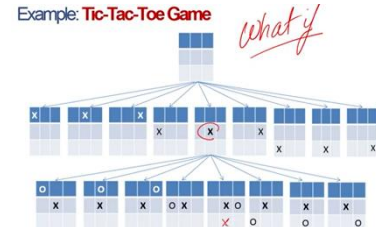
○ A Chess player considers a few alternative moves.

○ A mathematician chooses from a different strategies to find a proof for a theorem.

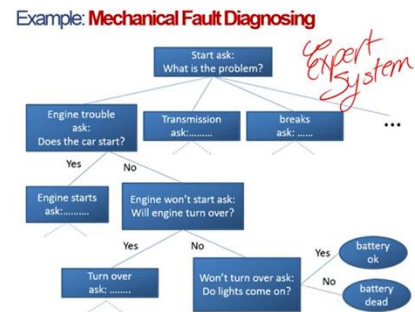
○ A physician evaluates several possible diagnoses.

هنا ال state space هي problems are solved by searching among alternative choices يعني عندنا مشكلة او مشاكل وبنطلعها حلول وبنختار الاحسن ك optimal solution يعني لو واحد بيحل مسالة مثلا على code forces او hacker rank بتبقى ليها اكثر من حل واكثر من time وبيتعملها اكثر من test فلما بيوصل لحل سريع وبسيط ده بيبقى ال optimal solution

عندك هنا ال x o عندك تسع أماكن وانك تجرب تحل كل المرات فتهطلع 9! فانت بتحاول تفكر انك تحل بانسب طريقة علشان تكسب على اللي قدامك اكيد مش هتقعد تحط الاحتمالات كلها علشان تحلها



ومثال تاني مثلا لو انت مركز صيانة عربية شوف العربية كام قطعة وكام قطعة ممكن تبوظ فانت لما بيجي الراجل بتسالو على اللي بيحصل بدل متجرب قطعة قطعة وتشوف المشكلة فين فالراجل لما يقول المشكلو فانت تقدر تميز مكان المشكلة واي اللي ممكن يتصلح من الأسئلة بس فهنا نقدر ان ال search مهم



لحل المشاكل وتبسيطها

How Human Beings Think..?

- طب مجربتش تسال نفسك هل انت عمرك لما دخلت مشكلة او جيت تلعب بصيت على كل الاحتمالات ولا بيبقى عندك خبرة فيستخدمها علطول علشان تحل المشكلة لان انت لو هتبص على كل الاحتمالات ممكن اللعب متخلصش خاصة مثلا اللي بيلعبو شطرنج ولا دومنة فده بيبقى الاسمو heuristics search وده اللي نوعو inform search
- Human beings do not search the entire state space (exhaustive search).
 - Only alternatives that experience has shown to be effective are explored.
 - Human problem solving is based on judgmental rules that limit the exploration of search space to those portions of state space that seem somehow promising.
 - These judgmental rules are known as "heuristics".

Rules of Thumb

اه عندنا أنواع من السيرش هما نوعين

*Exhaustive
Uninformed
Brute Force* | *Heuristic
Informed*

١. Exhaustive\uninform\brute force

وده بقى يسيدي ببقى نوع متعب عمتا ال\ دي يعني او كلهم نفس الحاجة المهم ان النوع ده بيبقى مكلف وصعب وخاصة مع الحاجات الكبيرة لي بقى لان بيلف على كل الاحتمالات بس بييجيب ال optimal solution وده الميزة اللي فيها اما هو ال time بتاعو كتير وال cost بتاعو اكثر

٢. Heuristic/informed

ده النوع اللي بنفضلو لانو بيبقى اسرع واحسن لانو بيبقى معتمد على خبرتك او الحاجات اللي ممكن تلاقيها بسيطة يعني يعني زي مقولنا ال x o او ال chess ده مستحيل حد يفضل يفكر وزي مقولنا لو استخدم ال exhaustive وفضل يشوف الاحتمالات كلها اه هينجح بس بعد أي وقت كتير و تكلفة اكثر اما heuristic ده حلو بتشوف خبرتك او بتشوف مثلا الاستراتيجية او الحلول اللي هتوصلك بسرعة وبس كده وقد يكون optimal solution or nearby for optimal solution

Heuristic Search

- A heuristic is a strategy for selectively exploring the search space.
- It guides the search along lines that have a high probability of success.
- It employs knowledge about the nature of a problem to find a solution.
- It does not guarantee an optimal solution to the problem but can come close most of the time.
- Human beings use many heuristics in problem solving.

هنبدا بقى نعرف ال search وازاي نستخدمو مثلا فى بروبليم ف هنبدا بمثال بسيط

Search

عندنا دلوقتي goal-based agent الى

کنا اتکلما علیہ فی section 2.1 ده

يبقى لي هدف معين ومعروف الاوت بوت

بتاعو ان هييقي مثلا انو ييقي وصل زي

مثال الرووم اللي اتكلما عليه اما ال

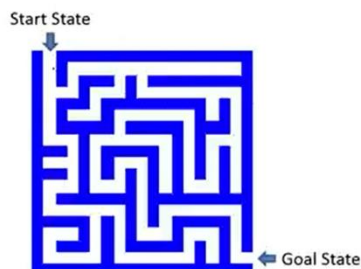
cost-based ده بیبقی مجهول انو

هيطلع الاوت بوت قد أي زي بتاع الايميل

مش عارفین ہیطلع کام ایمیل او هیمسح

کام ایمیل

We will consider the problem of designing **goal-based agents** in **fully observable, deterministic, discrete, known** environments.



وبعدین عندنا ف المثال ده ال environment type in section 2.2 فقولنا انو fully

observable وده معناها معروف عنو كل حاجة وده كل variable is known by

value وعندنا انو deterministic انو العامل الوحيد ف اتخاذ القرار ومفيش أي تدخل

عشوائي ولا في منافس يعني لا stochastic ولا strategic وانو discrete علشان القيم

معلومة ومش بتتغير علطول او ممكن نقول انها داخل sets و known لان القواعد معلومة

ان المفروض يدخل فين وهيخرج فين على عكس ال unknown

في حاجة بقى بتقسيلي انوده متهياً لانو يبقى ضمن ال solutions او اقرب لل optimal

solution وببقي من السيرش في الحلول واللى بيقسلى ده اسمو

وده بیقیس علی حالتین انو بییقی

وصول للهدف او انوال cost

خفيفة وهنا اقصد بال cost انو

time ,space ,cost of

The agent must find a *sequence of actions* that reaches the goal. The **performance measure** is defined by:

(a) reaching the goal, *and* ..

(b) how “expensive” the path to the goal is.

money

المهم ان البداية دي initial
state(s) وان في فعل بنخو
علشان نبدا نحل وده ال action
وفي حاجة بتشوف كنا فين وبقينا
فين دي ال transition mode

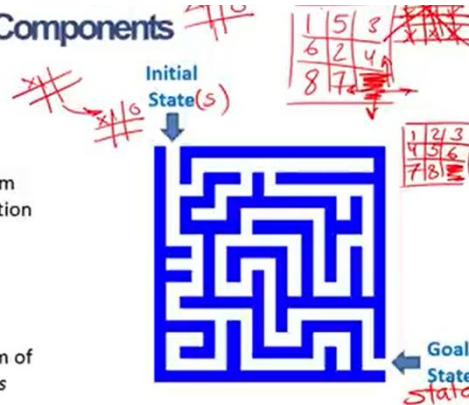
وفي عندنا الهدف اللي هنوصلو وهو goal state(s) واحنا بنعمل الحاجات دي في عداد
بيحسب اللي عملنا سواء كتايم او مسافة وهو path cost وبعد كده بيطلع معنا لما بنجرب
اكثر من مرة ونلاقي طريق احسن بنمشي عليه علطول وهو ال optimal solution the
best and lowest path for reaching the goal

يعني مثال لو انا مسافر من القاهرة (initial state) وقمت طالع على طريق اسكندرية الصحراوي مثلا فده (action) خدتو واتصل عليا صحبي سألني وصلت لفين ورديت عليه قولتلو عند اسكندرية كده فده (transition model) اللي هو طلعت من القاهرة وبقيت حاليا في الإسكندرية المهم ان كملت ووصلت الساحل(goal state) فقالي صحبي اده أي اللي اخرك فقولتلو انو الطريق كان ٧٠٠ كيلو ف تمن ساعات فده ال (path cost) فقالي مانت لو كنت جيت من طريق بنها الحر مثلا كان زمانك مشيت اقل وجيت اسرع فده ال (optimal solution) عمتا انا مروحتش الساحل قبل كده فاكيد معرفش المسافة او المشوار

romania نفس المثال بس انت بتحبو

- Initial state**
- Actions**
- Transition model**
 - What state results from performing a given action in each state?
- Goal state**
- Solution Path**
- Path cost**
 - Assume that it is a sum of nonnegative *step costs*

The **optimal solution** is the sequence of actions that gives the *lowest* path cost for reaching the goal.



Example: **Romania**

- On vacation in Romania; currently in Arad.
- Flight leaves tomorrow from Bucharest.

Initial state

- Ar

Actions

- Go from one city to another'''

Transition Model

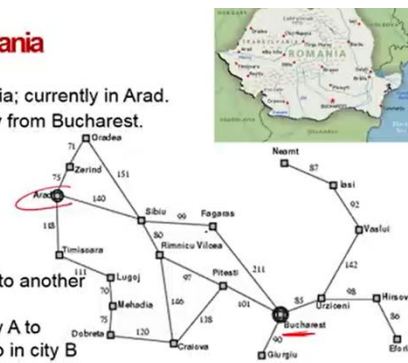
- If you go from city A to city B, you end up in city B

Goal State

- Bucharest

Path Cost

- Sum of edge costs (*total distance traveled*)



احنا كنا قولنا ال state space search دي بتبقى ان عندنا مشكلة وبنجبلها الحلول وبنختار منهم اللي ينفع optimal solution فهي بردو لو نيجي نبصلها هنلاقي ان لما يبقى معانا ال components بتاعت ال problem وال solutions فده يبقى state space search اللي هو the set of all states or solutions from initial state to goal state وقالك ممكن the state space in a directed graph by represent nodes and link

State Space

The initial state, actions, and transition model define the **state space** of the problem;

- The set of all states reachable from initial state by any sequence of actions.
- Can be represented as a **directed graph** where the nodes are states and links between nodes are actions.

شوية ال components لل graph او states space لان احنا قولنا state space can be represented in graph والمفاهيم دي المفروض وخذنها في ال algorithm بس نعيد عليها بسرعة كده لو في نوت او حاجة على ال node ده بيبقى او حاجة على ال node ده بيبقى labeled graph وعادة بيبقى

State Space

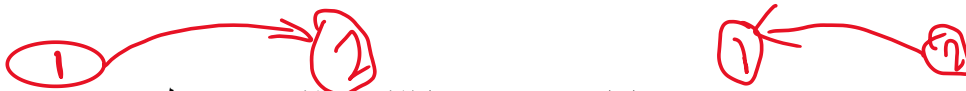
An AI problem can be represented as a **state space graph**.

A **graph** is a set of nodes and links that connect them.

Graph theory:

- | | |
|-------------------|---------------|
| ○ Labeled graph. | ○ Parent. |
| ○ Directed graph. | ○ Child. |
| ○ Path. | ○ Sibling. |
| ○ Rooted graph. | ○ Ancestor. |
| ○ Tree. | ○ Descendant. |

على ال edge وال edge or path ده اللي بيربط من node للتانية وعندك direct graph دي بتبقى ال edge متحدد طالعة من النود ولا داخلها



وعندنا ال undirect graph دي معناها مش مهم عندنا الاتجاه الاتنين متوصلين ببعض



دي معناها ان ممكن اروح من 1 to 2 والعكس

وعندنا ال rooted graph دي
 ساعات بتبقى موجود ومش شرط
 يعني بس هي لو موجودة بتبقى في
 الاغلب بتبقى بداية ال graph او
 ان ممكن أوصل منها لكل ال
 nodes والعكس وعادة بتبقى في
 ال tree

Tree دي مش حاجة ف الجراف
 بس ده concepts كبير وساعات

بنستخدم ال graph علشان نرسمو ال tree دي بتقى عادة
 labeled graph ,rooted graph,acyclic,undirected

في ال tree بقى شوية مفاهيم كده زي ال
 parent,child,sibling,ancestor,descendant

parent يعني ١ ل ٢ و ٣

طب وال ٢ و ٣ يعتبر children لل 1

ولو تبصلها انهم في نفس ال level فيبقى sibling و

لان two children node in same level is sibling

وهنا طلع group of parents of جديد

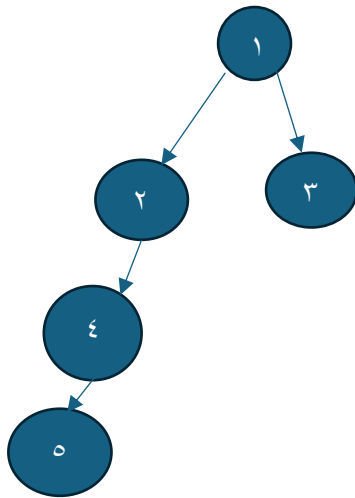
same node is called ancestor

يعني دلوقتي node ٤ ال ancestor بتاعها ١ و ٢

طب أي تاني ال group of children for one parent is

Called descendants

يعني دلوقتي مثلا ٢ ال descendants بتاعتها ٤ و ٥



State Space

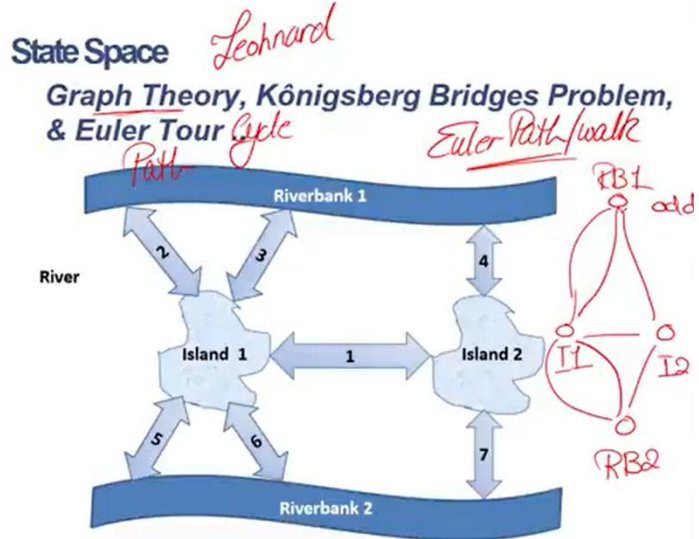
An AI problem can be represented as a state space graph.

A graph is a set of nodes and links that connect them.

Graph theory:

- Labeled graph.
- Directed graph.
- Path.
- Rooted graph.
- Tree.
- Parent.
- Child.
- Sibling.
- Ancestor.
- Descendant.

هنا كان مشروع وفشل بس كملت
معانا ال graph وهو هنا كان
عايز يعدي على كل الجسور من
غير ميعدي على واحد مرتين وان
يعرف يوصل لكل النقاط فقالك
ان ال euler walk انو
مينفعش يبقى عندنا في ال
graph تبقى odd ولما رسمها
لفي علشان يحقق المعادلة دي انو
يبقى كلو odd يبقى مش هتنفع
euler walk فجرب يشوف ال



euler tour وده بقى برودو منفعش لان كان تبقى cycle with even degree in each
node وبرودو منفعش وكملنا احنا بقى على ال graph

هنا فيما معناه يعني ان لو
المساحة او زودت الحاجات اللي
هيعملها ال search فطبيعي
ان state complexity واه
مش معناها ان path cost
هيزيد يعني زي لعبة ال
puzzle مهما كبرت اللعبة ف
الأول والآخر ان ال path
cost هيبقى واحد لان يدوبك
انت هتخط القطعة بس وانتبهنا

Example: the 8-Puzzle

States

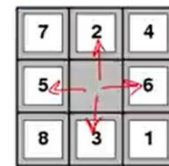
- Locations of tiles
 - 8-puzzle: 181,440 states ($9!/2$)
 - 15-puzzle: ~10 trillion states
 - 24-puzzle: $\sim 10^{25}$ states

Actions

- Move blank left, right, up, down

Path Cost

- 1 per move

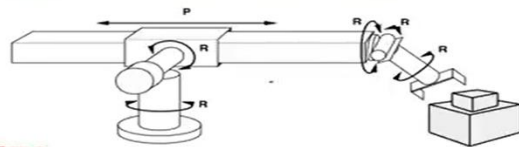


Start State



Goal State

Example: Robot Motion Planning



States

- Real-valued joint parameters (angles, displacements).

Actions

- Continuous motions of robot joints.

Goal State

- Configuration in which object is grasped.

Path Cost

- Time to execute, smoothness of path, etc.

اما تعالى بص هنا لو هنتشيل حاجة
بالجهاز ده فانت هتتحرك والمسافة هتتغير
يعني set space زيادة وتلف الدراع لو
مش معدول وتروح تحطوف ماكن تاني
وشوف هتنزلوف ارتفاع قد أي كل ده
path cost مختلف على حسب انت
هتعمل أي وهتودي من لفين لفين وهكذا

• دي ممكن اكثر حاجة اتكلمت فيها ورغي نوعا بس علشان تبقى فاهم الجزئية

دي مفاهيم قد تكون تكون صعبة من كمية الأرقام اللي شايفة اصل انا شايفة ومخضوض وقلقان بس تعالى متقلقش كلام بسيط وسهل اهو يعني ال state space complexity ده معناها انك بتحسب كل مكان ليه كام احتمال بمعني انت عندك x-o يتخط x او o يتسيبها فاضية وده معناها ان عندنا 3 اس 9 أماكن لما تيجي تبص للي شطرنج هما 6 اشكال بلونين فانت كانك عندك 12 شكل وفي احتمال يبقى فاضي

	Tic-Tac-Toe	chess	Go
State Space Complexity	$3^9 \approx 20,000$	$\approx 10^{40}$	$\approx 10^{170}$
game tree complexity (path)	$9! \approx 360,000$	$\approx 10^{120}$	$\approx 10^{360}$
plies (average game length)	9	70	150
Branching Factor	4	$\approx 31-35$	≈ 250

صحح معايا كده يعني هما 13 شكل او احتمال عندنا كم مربع عندنا 64 يبقى هي المفروض تبقى 13 اس 64 كلام مضبوط بس الشطرنج لعبة معقدة كل جزئية ليها أماكن معينة وقواعد وكده فلما بسطوها وحسبو حسبتهم طلعوها كده اللي هي عشرة اس 40

هنيجي لل game tree complexity دي لو فكرين ف سكشن 3.1 واحنا بنشرح السيرش على ال x-o فقولنا هنشوف بيرد علينا بايه واحتمالية الباقي ف الجراف فهنا نفس الكلام احنا كنا 9! لما انا لعبت صحبي بقي عندو 8! وبفضل تنقص بناء على الرد فهكذا حتى ف ال chess بس زي قولنا ان ال chess دايم صعبة ف حسبتها علشان قواعدا وعددها الكبير

نيجي لل plies دي زي منقول كده المتوسط للجيم بيخلص بعد كم حركة ف x-o ممكن يخلص في تسعة ومحدث يكسب او حد يكسب ممكن يخلص في خمسة والشطرنج ممكن حد يكسب بسرعة اقل من 70 حركة او بعد السبعين او اقل

وأخيرا ال branching factor دي برود المتوسط بتاع الجزء ده ممكن يتلعب كام مرة يعني انت يوم مهتمعظم انت وصحبك ف x-o ومحدث يكسب فهيقي في 5 اكس مثلا 4 o بس ف المتوسط أربعة وهكذا بقي هو الدكتور بيقول لو في tree or graph بتحسب ال average of degree for each node or child

some example for state space

Example: Traveling Salesperson

A salesperson has five cities to visit and then must return home.

- **Nodes(N)**: represent 5 cities.
- **Arcs(A)**: labeled with weight indicating the cost of traveling between connected cities.
- **Start state(S)**: a home city.
- **Goal states(GD)**: an entire path contains a complete circuit with minimum cost.
- **Complexity**: $(n-1)!$ Different cost-weighted paths can be generated.

Example: Traveling Salesperson

A salesperson has five cities to visit and then must return home.

- **Nodes(N)**: represent 5 cities.
- **Arcs(A)**: labeled with weight indicating the cost of traveling between connected cities.
- **Start state(S)**: a home city.
- **Goal states(GD)**: an entire path contains a complete circuit with minimum cost.
- **Complexity**: $(n-1)!$ Different cost-weighted paths can be generated.

و
عندنا strategies for state space search
عندك ال data-driven search
وده بنستخدمو لو data is given but the goal doesn't know form

State Space Search Strategies .. Selecting Search Strategy

Data-Driven Search is suggested if:

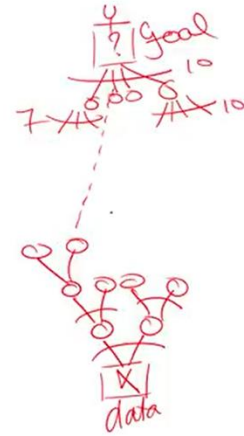
- The data are given in the initial problem statement.
- There are few ways to use the given facts.
- There are large number of potential goals.
- It is difficult to form a goal or hypothesis.

Goal-Driven Search is appropriate if:

- A goal is given in the problem statement or can easily be formulated.
- There are large number of rules to produce a new facts.
- Problem data are not given but acquired by the problem solver.

هنا بيبقى عندنا داتا بس مش عارفين هنوصل لايه وهنوصل للهدف ازاي فبنجمع الداتا كده ونعمل عليها اللوجيك والالجورزيم وده بيخلينا نوصل للهدف او نعمل احنا هدف جديد اما لو عندنا goal is given فاحنا فنفضل نحل ونجيب البراهين اللي وصلنا بيها للهدف ده وساعات بيبقى عندنا الهدف والداتا واحنا بنختار اللي ال strategies اللي هنمشي عليها زي المثال ده

يعني انا لو عايز احبيب تريخ حد من زمان اوي اللي هو يجي من ١٠٠ سنة مثلا هل اشوف من نحيتو لغاية ماوصل لعندي ولا العكس لما تيجي تبصلها هتلاقي ان كل شخص ليه اب وام فھفضل انزل بتاريخ الإباء وامهات اما لو جيت من عندو وقعدت اشوف عيالو وعيالو لغاية موصل لعندي الدنيا هتطول مني جامد



هنا هنبين الحاجات اللي خدناها في ال set space يعني كلام يعتبر مفهوش جديد اوو هنا بيقولك انو بندا من ال initial state

Search: Basic idea

وبعدين

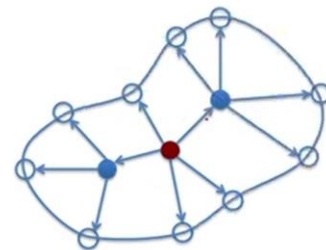
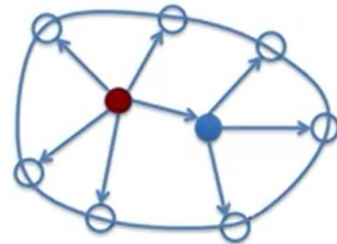
Start

Search: Basic idea

ولما راح عند كل الاحتمالات اللي عندو ال graph بيتعمل expanded وهنا كل node وال adjancet node بتوعها حاجة اسمها frontier او ال fringe فطبعا ده بيزيد معنا وهنعرف اكثر عن الكلام ده بعد كه في ال algorithm search like



el back tracking, Depth first search,



طبيعي ان ال frontier او ال fringe بيبدأ من initial state وبيزيد كل ماروح عند node تانية وعلى حسب عدد ال nodes ب expanded

وبنختار ال search strategy اللي خدناها قبل كده وهي heuristic or exhaustive

وبنشوف ال node اللي وصلناها دي goal ولا لا لو

goal خلاص بقفل ال search ويبقى معايا solution لان مش احنا قولنا ال solution is

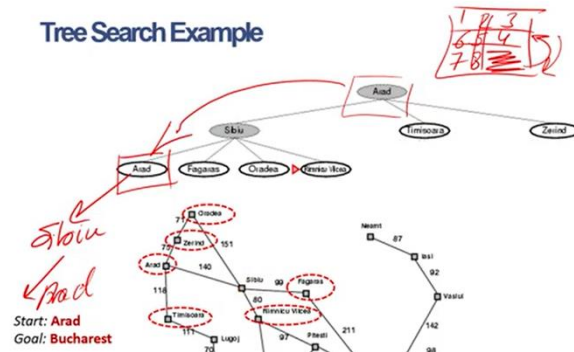
Tree Search Algorithm Outline

- Initialize the frontier using the starting state. *fringe*
- While the frontier is not empty:
 - Choose a frontier node according to search strategy. *Blind Exhaustive Heuristic Informed*
 - and take it off the frontier.
 - If the node contains the goal state, return solution. *Solution Path*
 - Else expand the node and add its children to the frontier.

A path or paths from initial state to goal state

طَبُّهُ لَوْ مَشِىَ **goal** هَيَفُضْلُ يَتَعَمَلُو **expanded** لَغَايَةُ مَنَاقِشِ حَاجَةُ فَالِ **fringe**
وَسَاعَتُهَا مُمْكِنُ نَقُولُ أَنَّ **goal** مَشِىَ مَوْجُودٌ مِثْلًا

Tree Search Example



هنا اللي قولناه فوق هما بيعمل check ان كانت روحتهلا قبل كده ولا لا في حاجة اسمها explored set او deadblock end او ممكن يسميها ك visited المهم انها بتتشوف اللي انها حد زارها قبل كده ولا لا أحيانا لو وصلها اكثر من طريق بيخدو الطريق الأقل ويمسح والثاني

Handling Repeated States

- Initialize the **frontier** using the **starting state**
- While the frontier is not empty
 - Choose a frontier node according to **search strategy** and take it off the frontier.
 - If the node contains the **goal state**, return solution
- Else **expand** the node and add its children to the frontier.

To handle repeated states:

Every time you expand a node, add that state to the **explored set**; do not put explored states on the frontier again.

Every time you add a node to the frontier, check whether it already exists with a higher path cost, and if yes, replace that node with the new one.

هنا هنبدأ نتكلم عن ال back tracking ومن تعريفو اللي is a technique for systematically trying **all paths through a state space**

هنا اقدر اقولك انو **brute force** او **uninformed** ,**exhaustive**

Backtracking Search

Brute force Blind Exhaustive Uninformed
 "Backtracking is a technique for systematically trying all paths through a state space"

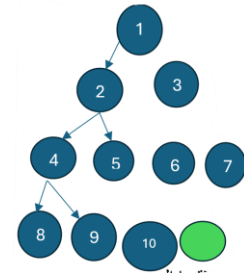
It begins at the start state and pursues a path until:

- Finding a goal, then quit and return the solution path.
- Finding a dead end, then backtrack to the most recent unexamined node and continue down one of its branches.

لانو بيعدي على ال path ولو لقي ال goal node or state بيطلع

بس أسلوب ال back tracking ازاى بقى هو عمنا شبه ال depth first search فطريقة التحرك بتاعتو بس مختلف ف ال data structure متنساش ان depth first انو بيساعد ف ال optimal solution

هنا ال back tracking بيمشي ازاى بقى قولنا بنحط ال fringe بدايتو من ال initial



current state (CS) اللي هي رقم ١ بيمشي ال back tracking من الشمال

يخلصو وبعد كده يخش على اليمين يعني هو هينزل على 2 هل هي ال goal

ده بيتسيف فين بقى على state list طب أربعة وخمسة دول بالنسبة لنا أي انها

New state list (NSL) هيخش على 4 مش هي ال goal هيخش على 8 مش هيلاقيها ال

goal فيقوم عامل back وتبقى دي DEAD END

بقى على 4 ويطلع منها على 9 ومش هيلاقي انها ال goal

فهيطلع من التسعة ل 4 ويخش على ٢ مش هيلاقي انها ال GOAL ويقوم داخل على ال 5

ويرجع ثاني على ال 2 ويطلع على الواحد يخش على ال 3 ويعمل نفس اللي عملو النحية الثانية

