Search 2: Strategies	start time:

This activity introduces some ideas and techniques used in artificial intelligence (AI) to search for possible solutions to problems.

Before you start, complete the form below to assign a role to each member.
If you have 3 people, combine Speaker & Reflector.

Team	**Date**
Team Roles	**Team Member**
Recorder: records all answers & questions, and provides copies to team & facilitator.	
Speaker: talks to facilitator and other teams.	
Manager: keeps track of time and makes sure everyone contributes appropriately.	
Reflector: considers how the team could work and learn more effectively.	

This activity uses several icons:

🔑 is a **key** question; you should have a good answer that everyone understands.
❓ is an **optional** question; you should skip it if you are behind schedule.
✋ is a **pause**; check with the instructor before you continue.
🛑 is a **stop**; wait for other teams to catch up before you continue.

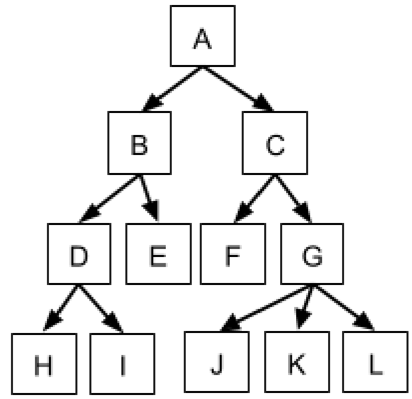
Reminders:

1. *Note the time whenever your team starts a new section or question.*
2. *Write legibly & neatly so that everyone can read & understand your responses.*

It's so much easier to suggest solutions when you don't know too much about the problem.
-- Malcolm Forbes

(16 min) A. Search Structure	start time:

1. (2 min) During search, some of search space has been **explored** and the rest is **unexplored**. The explored part can be represented as a **tree** with a **node** for each state that has been explored, an **edge** (arrow) for the action between two states, and where the **initial state** is the **root** (top) node of the tree. (Many search algorithms use tree-based data structures.)



a.	Which state is the **initial state**?	
b.	Which states are one step (action) away from the initial state?	
c.	How many actions are needed to get from state A to state J?	

2. (2 min) The border between **explored** and **unexplored** space is the **frontier**.

a.	Which states are on the **frontier**?	
b.	How do frontier states differ from non-frontier states?	

3. (2 min) If a typical node (including the root) has **2** possible actions:

a.	How many nodes are 1 action away from the root?	
b.	How many nodes are 2 actions away from the root?	
c.	How many nodes are 3 actions away from the root?	

4. (2 min) If a typical node (including the root) has **4** possible actions:

a.	How many nodes are 1 action away from the root?	
b.	How many nodes are 2 actions away from the root?	
c.	How many nodes are 3 actions away from the root?	

```
1: Add the initial state to the (empty) frontier.
2: If the frontier is empty, STOP (FAILURE).
3: Choose and remove one state from the frontier.
4: If that state is the goal state, STOP (SUCCESS).
5: Add that state's neighbors to the frontier.
6: Go back to step 2.
```

5. (4 min) Many search algorithms use the six steps shown above. Which step(s):

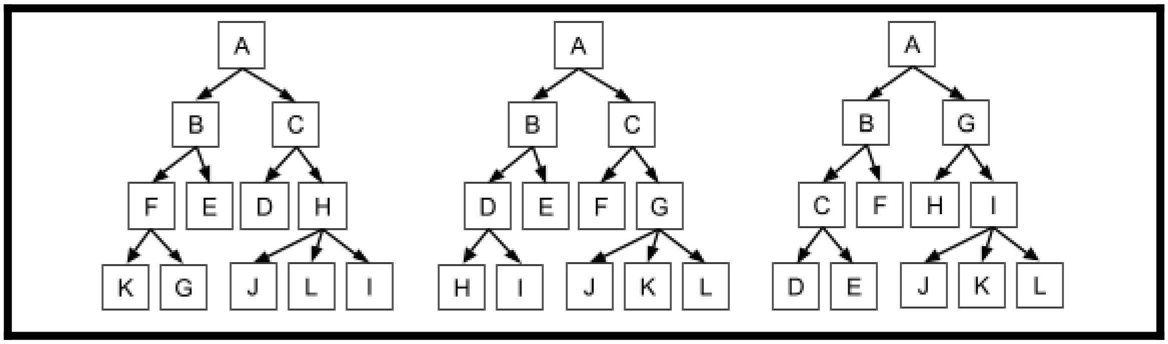
a.	make the frontier **bigger**?	
b.	make the frontier **smaller**?	
c.	end the **search**?	

6. (2 min) 🔑 Explain why the frontier's size and shape is an important aspect of search.

7. (2 min) 🔑 A search that explores all of state space is a **complete** or **exhaustive** search. Why might **exhaustive search** not be practical?

✋ Check with the instructor before you continue.

(10 min) B. Search Strategies	start time:



1. (3 min) The six steps above describe when to add and remove frontier states,
but not how to choose which state to visit next. This can be done in several ways.
In the diagrams above, the letters show the order in which states were visited, starting with A.
In **column 1** below, choose the diagram (*left*, *middle*, *right*) that matches each description.

	Description	1. Diagram	2. Name
a.	Visit all states that are 1 edge away, then all states that are 2 edges away, etc.		
b.	Follow child states as deeply as possible. When the search reaches a state without children, it backtracks to an unvisited child and continues.		
c.	Choose any frontier state at random.		

2. (2 min) These 3 strategies are called **random-first**, **depth-first**, and **breadth-first**.
In **column 2** above, choose the name that best matches each description.

3. (4 min) Of these 3 strategies, decide which one(s):

a.	would find the solution with the minimum path length first?	
b.	would be useful for experimentation, but not very practical?	
c.	would be likely to find solutions with very long paths, before solutions with shorter paths?	
d.	would tend to stay in nearby areas of the frontier, rather than jump between different parts of the frontier?	

✋ Check with the instructor before you continue.

(10 min) C. Search Variations	start time:

1. (2 min) If **depth-first search** is limited to a **maximum path length**, it is called **depth-limited search**. (Recall that **path length** is the number of actions on a path.)

a.	What is an advantage of depth-limited search?	
b.	What might go wrong if the max path length is too **small**?	
c.	What might go wrong if the max path length is too **large**?	

2. (2 min) 🔑 The **path cost function** (see earlier activity) could return the **path length**, but it could be different if states or actions could have different costs
(e.g., in Traveling Salesperson, the cost to visit a city, or the distance between cities).
Explain how **breadth-first search** might not return the minimum cost path.

3. (4 min) 🔑 Another strategy could balance **path cost** instead of **path length**.
Note: this is **different** from always choosing the next action or state with the lowest cost.
Describe how this **uniform-cost search** works.

4. (2 min) ❓ To save time, search tries to avoid states that can be excluded easily.
Techniques to ignore or remove tree branches for efficiency are called **pruning**.
For example, a search could keep a list of every state it visits.

a.	How could this list save time?	
b.	What problem could arise in a very large state space?	

✋ Check with the instructor before you continue.

(14 min) D. Search Direction	start time:

In a **word ladder** puzzle, the goal is to find a sequence of real words, from a **start word** to an **end word**, where each word differs by just one letter from the previous one. For example, a word ladder from PATH to MIST is:

PATH ⇒ MATH ⇒ MASH ⇒ MAST ⇒ MIST

1. (2 min) Find a word ladder from COLD to WARM.

2. (2 min) In a word ladder:

a.	What is the **initial state**?	
b.	What is the **goal state**?	
c.	Does the structure of the puzzle change if we swap the initial and goal states?	
d.	How many letters are in the English alphabet?	
e.	How many 4-letter sequences are possible?	
f.	There are ~5000 4-letter words in English. What percentage of 4-letter sequences are words?	

3. (3 min) Assume that every state has applicable actions that lead to 2 new states. Starting from the initial state, how many states are:

a.	**1 action** away?	2	e.	within **1 action**?	3 = 1 + 2
b.	**2 actions** away?		f.	within **2 actions**?	
c.	**3 actions** away?		g.	within **3 actions**?	
d.	**N actions** away?		h.	within **N actions**?	

		8	7	6	5	4	3	2	3
	8	7	6	5	4	3	2	1	2
8	7	6	5	4	3	2	1	G	1
	N	7	6	5	4	3	2	1	2
		8	7	6	5	4	3	2	3

4	3	4				4	3	2	3
3	2	3	4			3	2	1	2
2	1	2	3	4	3	2	1	G	1
1	N	1	2	3	4	3	2	1	2
2	1	2	3	4		4	3	2	3

4	3	4	5	6	7	8			
3	2	3	4	5	6	7	8		
2	1	2	3	4	5	6	7	G	
1	N	1	2	3	4	5	6	7	8
2	1	2	3	4	5	6	7	8	

4. (2 min) 3 state space diagrams are shown above. Which one (*left, middle, right*) shows:

a.	path length from the initial state N?	
b.	path length from the goal state G?	
c.	path length from the initial or goal states?	

5. (2 min) 🔑 Explain why it could be much more efficient to search from the initial and goal states at the same time - this is **bi-directional search**.

6. (2 min) Which of these search strategies could use **bi-directional search**?

a.	breadth-first search	
b.	depth-first search	
c.	uniform-cost search	

🛑 Wait until the instructor tells you to continue.

(20 min) E. Informed Search	start time:

Uniform-cost, breadth-first, and depth-first search **don't know** if a path is a goal path until a goal state is reached; thus, these strategies are called **blind** or **uninformed**. Uniform-cost search chooses frontier states using an **evaluation function** which is often the **path cost function**, based on a path's length, states, and actions. Search can be better if the evaluation function tries to **predict** good paths, even if such predictions are not perfect. There are called **heuristic functions**, and these search strategies are called **informed** or **heuristic search**.

I.

1	2	3
4	5	6
7	8	

II.

2	3	6
1	5	
4	7	8

III.

1	2	6
8	7	4
3	5	

IV.

1	2	
4	5	6
3	8	7

V.

1	4	6
7	2	5
	8	3

1. (3 min) The figure above shows 5 states for an **8-puzzle**. State I (left) is the **goal state**.

Answer each question below for each state.		I.	II.	III.	IV.	V.
a.	Does the state match the goal state?	Y	N			
b.	How many of the 8 tiles are in the wrong position?	0				

2. (4 min) Refer to your answers above for the questions below.

a.	Could **1a** be a good **goal test** (to detect a goal state)?	
b.	Could **1b** be a good **goal test**?	
c.	Could **1a** be a good **heuristic** to find states **near** the goal?	
d.	Could **1b** be a good **heuristic** to find states **near** the goal?	
e.	Which state (II or IV) is **easier to solve** (closer to the goal)?	
f.	Would **1b** be a useful heuristic to compare these 2 states?	

3. (2 min) Explain why **1a** and **1b** are not ideal heuristics.

frontier state:	**J**	**K**	**L**	**M**	**N**
cost & heuristic values:	c=2 h=8	c=10 h=2	c=4 h=4	c=8 h=2	c=2 h=10

4. (3 min) The table above lists 5 states (J...N) on the frontier, each with a **cost c** (from the **initial** state) and **heuristic h** (estimated cost to reach a **goal state**).

a.	Which states should be searched next, **based only on c**?	
b.	Of these, which one should be next, **based on h**?	
c.	Which states should be searched next, **based only on h**?	
d.	Of these, which one should be next, **based on c**?	
e.	Which states should be next, **based on both c and h**?	

5. (4 min) 🔑 Describe how to use a heuristic function in an evaluation function to improve (blind) **uniform-cost search**. The result is called **best-first search**.

6. (2 min) Describe at least one other **heuristic function** that could used for **8-puzzles**.

7. (2 min) ❓ Describe at least one useful **heuristic function** that could be calculated from a state for **8-queens** or **normal magic square** (your choice),

There are many types of heuristic functions, and multiple ways to use them to improve search. We might explore some of them later in the course.

🛑 Wait until the instructor tells you to continue.

(7 min) F. Pseudocode	start time:

```
def search( problem ):
  # a.
  frontier = makeEmptyList()
  frontier.add( problem.getInitialState() )
  # b.
  while ( ! frontier.isEmpty() ):
    # c.
    nextState = frontier.removeNext()
    if ( problem.isGoalState( nextState ) ):
      return nextState
    # d.
    neighbors = problem.getNeighbors( nextState )
    frontier.addAll( neighbors )
  # e.
  return None
```

1. (5 min) The pseudocode above shows the structure of many search algorithms. Match each letter above with the appropriate comment below.

`add neighbors to frontier and repeat`	
`get next state from frontier; if goal is found, return it`	
`if frontier is empty, search fails`	
`repeat until frontier is empty`	
`start with a frontier containing just the initial state`	

2. (2 min) Of the 3 search strategies (breadth-first, depth-first, random-first), which one(s):

e.	would most likely use a **queue** (first-in, first-out)?	
f.	would most likely use a **stack** (last-in, first-out)?	