# Search 3: Applications of Searching Algorithms	start time:

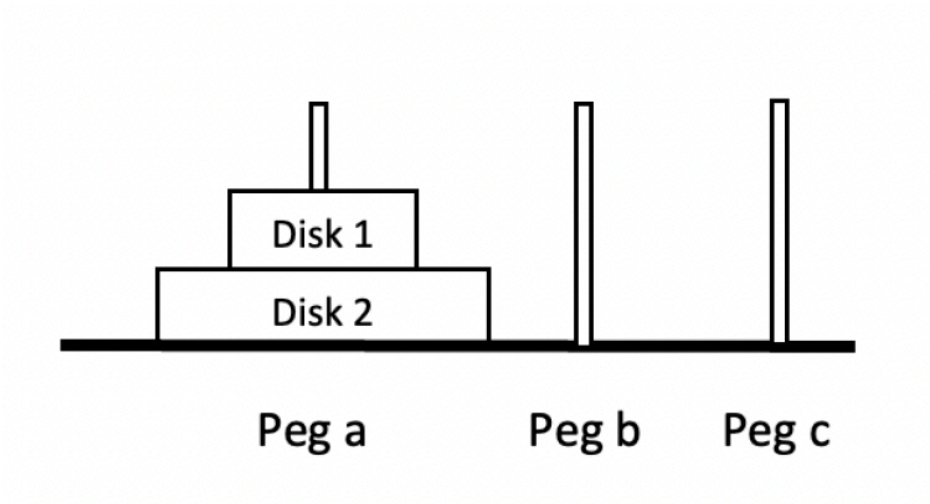
This activity introduces some ideas and techniques used in artificial intelligence (AI) to search for possible solutions to problems.

Before you start, share this document with your team member(s) and then complete the form below to assign the role of speaker.

Team Role	Team Member
Speaker: shares your team's ideas with the class.	Lauren, Arogya, Makenna

Tower of Hanoi

Part A. State Space	start time:



In the classical Tower of Hanoi game, there are three pegs: A, B, and C, along with *n* disks of varying diameters. The disks are numbered in decreasing order, meaning the smallest disk is labeled as disk 1. Initially, all the disks are stacked on one peg, such as peg A, in decreasing order of size. The objective is to move all the disks, one by one, to another peg, such as peg C, while following the rule that a larger disk cannot be placed on top of a smaller disk at any time.

This game has been solved using recursive and iterative algorithms in CS1/2. We now work on this game again using graph search algorithms.

1. We define a state as a representation of the arrangement of all disks on three pegs.

The number of valid states for the game, aka we can only have larger disks under smaller disks

1.1 What is the size of the state space (number of different states) if there is only one disk in the game?

3, because there are three pegs and one disk

1.2 What is the size of the state space if there are two disks?

3 ways to put single disks on the three pegs (times by 2 because there are 2 disks), 3 ways to stack the disks on a single peg (9)

1.3 What is the size of the state space if there are three disks?

3*3*3

1.4 What is the size of the state space if there are N disks? (Hint: look at the pattern.)

3^N induction proof

2. We enclose everything we need for describing the state in a state representation. So we later on code the state in python.

Dictionaries and arrays (matrix)

2.1 What did we use to represent the state in eight puzzle in 306?

Tree

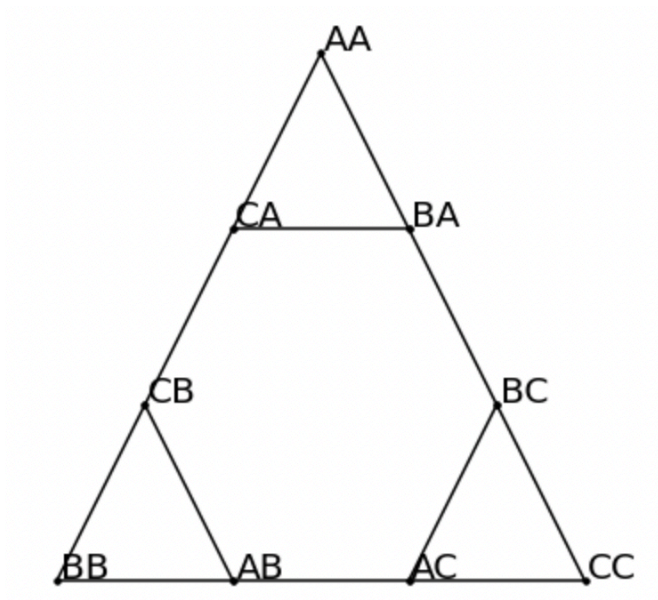
2.2 What did we use to represent the state in maze puzzle in 306?

Tree

2.3 Based on the answer in 1.4, propose a state representation of this game. The state describes the arrangement of all disks on three pegs. (Hint: think about permutation).

Dictionary of list:
{1:[...],2:[...],3:[...]}

Part B. States	start time:



3. Suppose we use a string of letters to represent a state. The ith letter of the string representation means the position of the ith disk.

3.1 How many disks are in the representation AA?

2

3.2 What does AA represent in the arrangement of the disks?

Peg A has all the disks

3.3 What does BB represent in the arrangement of the disks?

Peg B has all the disks

3.4 What does CA represent in the arrangement of the disks?

Peg C has disk 1 and peg A has disk 2

3.5 Write the state where there are 5 disks in total, the first three disks are on peg A, and the other two disks are on peg C.

AAA CC

4. What is the start state in the picture above?

AA

5. What is the goal state in the picture above?

CC

6. In general, given N disks, what is the start state, and what is the goal state?

Start state- All on A so N number of A
Goal state- All on C so N number of C

7. The states in the picture above are connected by edges.

Possible moves we can make from that state

7.1 What is the meaning of the edge between AA and CA?

Disk 1 is moved from peg A to peg C

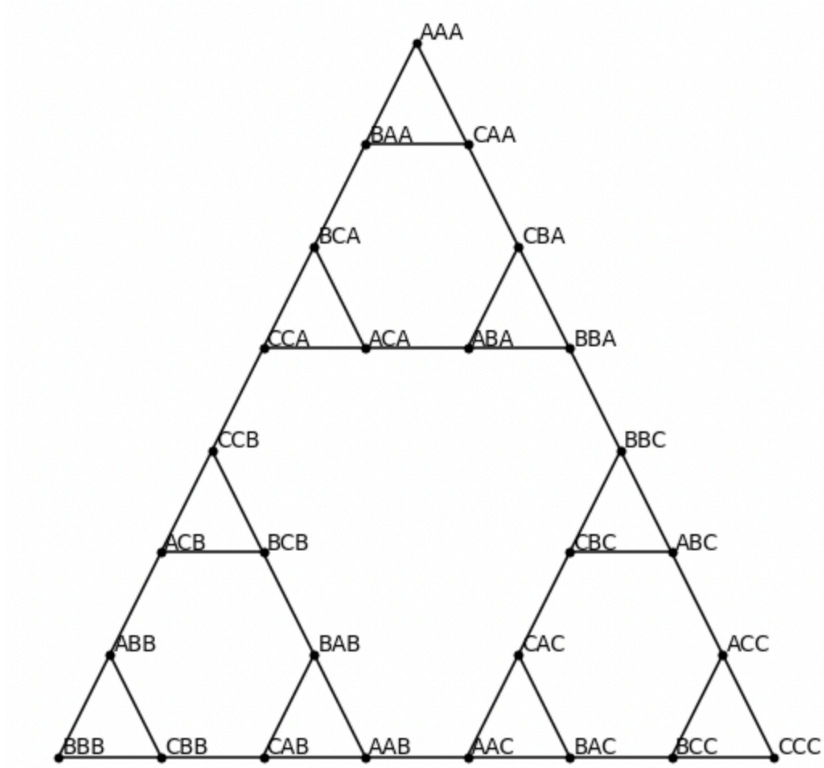
7.2 What is the meaning of the edge between BA and BC?

Disk 2 is moved from Peg A to Peg C

7.3 In general, what does the edge represent?

A disk movement

Part C. Actions	start time:



8. We can change states from one state to another state by moving only one disk from one peg to another peg. We call this one-time moving of disk an action. The graphical representation of an action is an edge in the graph.

8.1 In the pictures above, what is the minimum number of actions of a state?

2

8.2 In the pictures above, what is the maximum number of actions of a state?

3

8.3 When does a state have the minimum number of actions?

When all disks on one peg

8.4 When does a state have the maximum number of actions?

When more pegs have smaller disk and other one has bigger. We have possibility to move more than one disk to that position.

8.5 Is your conclusions above still true for N disks?

Yes because disks can only mobe to adjacent pegs

9. Executing an action on a state results in a new state. We call all new states resulting from one state as its successors.

9.1 List all successors of AAA using the above pictures. What actions are applied to AAA?

BAA CAA; the small disk can be moved from A to B or C

9.2 List all successors of CCB using the above pictures. What actions are applied to CCB?

ACB BCB CCA; small disk from C moved to A or B or largest disk from B moved to A

9.3 List all successors of CBC using the above pictures. What actions are applied to CBC?

CAC ABC BBC; the small disk from C moved to A or B or the small disk from B moved to A

9.3 In general, given a state with N disks, how do you find out the states of its successors? Summarize it in 2-3 rules below:

Switch first letter to the other two letters (guarantee 2 children)
Switch the first different letter to the the other peg (the letter you haven't seen before)

9.4 List all successors of

- AAAAA

  BAAAA
  CAAAA

- AAAB

  BAAB
  CAAB
  AAAC

- BBAC

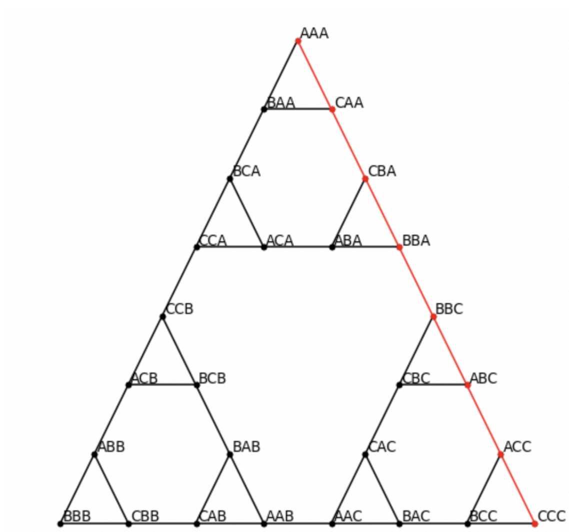
  ABAC
  CBAC
  BBCC

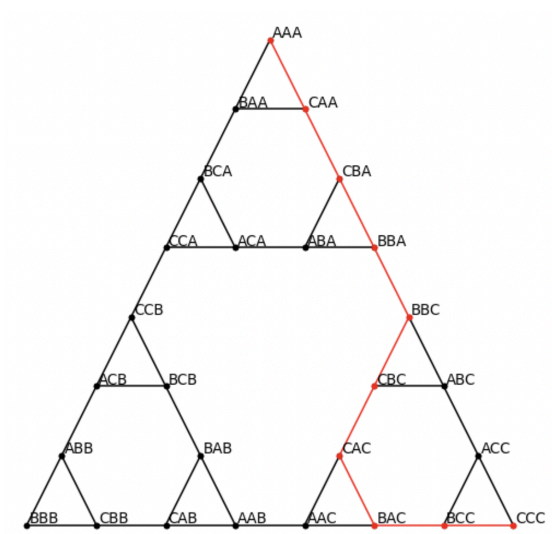
- CBACBBCCBBA

  ABACBBCCBBA
  BBACBBCCBBA
  CAACBBCCBBA

10. Look at the code now. You need to write python code to implement *def actions(self, state)* and *def result(self, state, action)* for what we just discussed in this mode.

Part D. Searching Algorithms	start time:





11. In the class Hanoi, how do you implement *goal_test(self, state)?*

Check if all letters in string are C

12. The path from start state to goal state forms a solution of actions. If moving one disk costs a constant value, which path (colored in red) in the pictures above costs less?

Left

13. One path is found by DFS and another one is found by BFS. Which path is found by which algorithm? Why?

Left BFS Right DFS

14. Look at the util.py, which class should you use to implement DFS or BFS? You are recommended to use the data structures defined in util.py in your assignment.

DFS - stack
BFS - queue (not priority because all moves cost the same)

15. Is DFS complete? Is DFS optimal? (complete - guarantee to find the solution if it exists. optimal - guarantee to find the optimal solution.)

DFS is complete but not optimal

16. Is BFS complete? Is BFS optimal?

BFS is optimal because it took the shortest path

17. If we use A* to implement the search, please design at least two heuristic functions below.

Count the number of letters not C in state, take the path with more C
Otherwise, the path with the higher weight

If largest disk not on goal peg then count the number of disks on top of the largest disk (number of disks * 2 -1 moves)
If largest disk on goal peg, then do it for the next largest disk

18. Are the heuristic functions you designed admissible? (admissible - the heuristic value of a state (the estimated cost from the state to the goal) is less than the actual cost to the goal.)

Yes, it is less than the actual cost to the goal.

19. Which heuristic function you designed is better in terms of optimality and efficiency? Why?

Both are giving the same optimization level.

19.1 Optimality (guarantee to finding the optimal solution)

Both same level optimality

19.2 Efficiency (the number of expanded nodes during the search)

Same path followed, so name number of expanded nodes

20 Suppose the heuristic function you designed is not admissible. List at least one situation that it still might find the optimal solutions. (This question is quite challenging!)

Our heuristic follows BBC to CBC but the most optimal is BBC to ABC

21. Look at the util.py, which class should you use to implement A*?

Priority queue

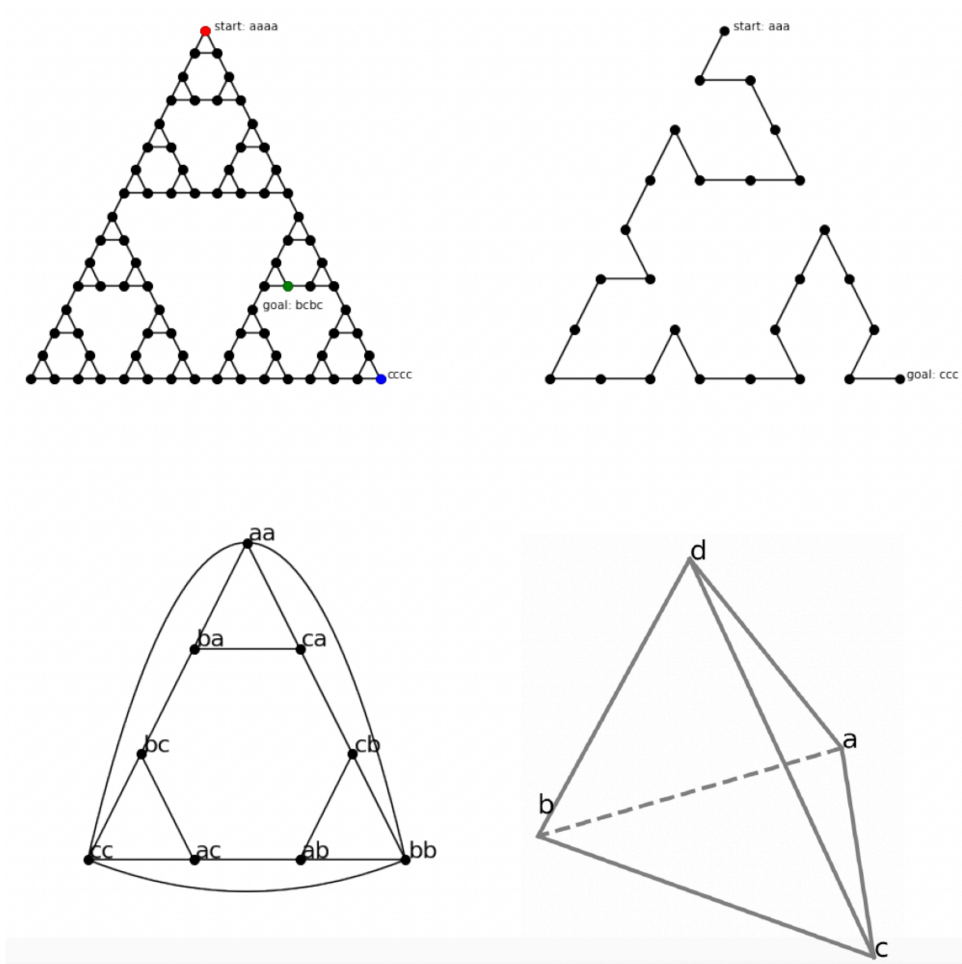
22. What is the shortest path giving N disks?

$2^N - 1$

23. What is the longest non-repetitive path giving N disks?

$3^N - 1$

Part E.	start time:



24. There are many variants of the tower of hanoi. They also can be solved using the graphical representation. Please match which graph represents which variants below.

- Move two alternative colors of disks to two pegs in same color - 3rd diagram bottom left
- Only move to adjacent peg(s) - 2nd diagram top right
- Four pegs - 1st diagram top left
- Move up to 2 stacked disks - 4th diagram bottom right