Practical 02
Algorithms

1. Khan Academy - Algorithms (Intro to Algorithms)

Background: *Khan Academy offers practice exercises, instructional videos, and a personalized learning dashboard that empower learners to study at their own pace in and outside of the classroom. We tackle math, science, computer programming, history, art history, economics, and more. Our math missions guide learners from kindergarten to calculus using state-of-the-art, adaptive technology that identifies strengths and learning gaps. We've also partnered with institutions like NASA, The Museum of Modern Art, The California Academy of Sciences, and MIT to offer specialized content.*

Khan Academy has partnered with Dartmouth college professors Tom Cormen and Devin Balkcom to teach introductory computer science algorithms, including searching, sorting, recursion, and graph theory. Learn with a combination of articles, visualizations, quizzes, and coding challenges.

a. Visit the following website (*https://www.khanacademy.org/computing/computer-science/algorithms#intro-to-algorithms*) to access Khan Academy's online instructional content on Algorithms (Intro to Algorithms). Go through the following instructional video and activities from the website:

- What is an algorithm and why should you care?
- A guessing game
- Route-finding
- Discuss: Algorithms in your life

b. With reference to the Khan Academy's instructional content on Algorithms (Intro to Algorithms), answer the following questions:

i. What algorithm is used by computer scientists to write checkers programs that never lose?

ii. Which data structure is used in the implementation of that algorithm?

iii. What are the 2 important criteria that a good algorithm should have?

iv. What technique is used by computer scientists to compare the efficiency of algorithms independent of the programming language or hardware used.

v. In the "Guessing game" activity, what are the 2 approaches (for guessing the number) demonstrated?

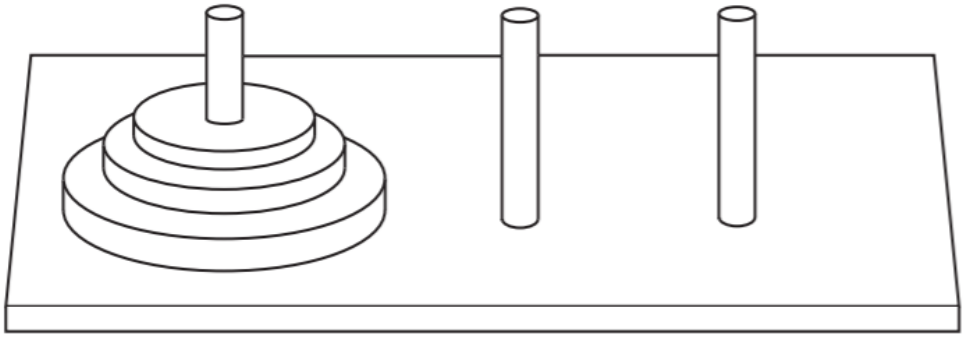
vi. In the "Guessing game" activity (and using the more efficient approach), guessing a number from 1 to 300 should not need more than 9 guesses. Why is that so?

vii. In the "Route-finding" activity, what is the algorithm used to find shortest path from Start to Goal in the maze?

2. Towers of Hanoi

The Towers of Hanoi is a mathematical game that is often used in computer science to illustrate the use of algorithm for problem solving, particularly the concept and power of recursion (which we will discuss further in Topic 05).

The game uses three pegs and a set of discs with holes through their centers. The discs are stacked on the leftmost peg, in order of size with the largest disc at the bottom.



The objective of the game is to move all the discs from the first peg to the third peg, using the middle peg as a temporary holder – with the minimum number of moves. In addition, the following rules must be observed while moving the discs:

(1) Only one disk may be moved at a time
(2) A disk cannot be placed on top of a smaller disc
(3) All discs must be stored on a peg except while being moved

(Content adapted from: Starting out with Python. Tony Gaddis, Addison Wesley, 2nd Edition, 2012.)

In Tutorial 02 – Question 3, you are asked to devise an algorithm to solve the Towers of Hanoi problem. Your tutor will discuss with you a possible algorithm (during the tutorial session) to solve the Towers of Hanoi problem. Based on this algorithm, an incomplete implementation of the solution for a Towers of Hanoi game with 3 discs is provided below.

Complete the implementation by providing the rest of the required codes.

A sample output of the completed program for a Towers of Hanoi game with 3 discs is as follows:

```
Move a disc from peg 1 to peg 3
Move a disc from peg 1 to peg 2
Move a disc from peg 3 to peg 2
Move a disc from peg 1 to peg 3
Move a disc from peg 2 to peg 1
Move a disc from peg 2 to peg 3
Move a disc from peg 1 to peg 3
All the pegs are moved!

Process finished with exit code 0
```

```
# This program simulates the Towers of Hanoi game.

def main():
    # Set up some initial values
    num_discs = 3
    from_peg = 1
    to_peg = 3
    temp_peg = 2

    # Play the game
    move_discs(num_discs, from_peg, to_peg, temp_peg)
    print('All the pegs are moved!')

# The move_discs function displays a disc move in
# the Towers of Hanoi game.
# The paramenters are:
#   num:        The number of discs to move.
#   from_peg:   The peg to move from.
#   to_peg:     The peg to move to.
#   temp_peg:   The temporary peg.
def move_discs(num, from_peg, to_peg, temp_peg):
    if num > 0:
        move_discs(________, ________, ________, ________)
        print('Move a disc from peg', ________, 'to peg', ________)
        move_discs(________, ________, ________, ________)

# Call the main function
main()
```

-- End of Practical --