**DSA Adventure Quest**

## 1. Introduction:

The **DSA Adventure Quest** is an interactive Python-based game designed to help users learn and practice Data Structures and Algorithms (DSA) concepts. The game consists of multiple levels where players must solve puzzles related to arrays, sorting algorithms, and graph connectivity to advance to the next level. The game is designed using the **tkinter** library, which provides a simple GUI interface.

## 2. Requirements:

### Software Requirements:

- **Python:** This game is written in Python 3.x. Ensure that Python 3.x is installed on your system.
- **tkinter:** This library is used to create the graphical user interface (GUI) for the game. It comes pre-installed with Python, so you do not need to install it separately.

### Hardware Requirements:

- A computer with basic processing power and a display that supports graphical applications**.**

## 3. Software Used and Technology:

### 1. Python:

The game is built using Python 3, a high-level programming language known for its simplicity and readability. Python's versatility makes it an excellent choice for rapid prototyping and GUI-based applications.

### 2. tkinter:

tkinter is the standard Python interface to the Tk GUI toolkit. It provides various widgets like buttons, labels, text fields, and more, which are used to create interactive user interfaces.

- **Widgets used in the game:**
    - Labels: To display text, instructions, and puzzle details.
    - Buttons: To trigger game actions such as starting the game, submitting answers, and progressing through levels.
    - Entry Fields: To allow user input for solving puzzles.

**3. Random Module:**

The random module is used to generate random numbers for puzzles, such as creating random arrays and graph structures for the game levels.

**4. Messagebox:**

messagebox from tkinter is used to display pop-up messages that inform the player whether their answer is correct or incorrect, providing immediate feedback.

**4. Game Design and Levels:**

The game is divided into three levels, each corresponding to a different DSA concept:

Level 1: Array Puzzle

- Objective: The player is given a list of random numbers and is asked to rearrange them to form the largest possible number.
- Puzzle Logic: The game generates a random array of 5 integers and prompts the player to rearrange them in descending order. If the player inputs the correct order, they move on to the next level.

Example:

    - Given numbers: [42, 5, 99, 15, 73]
    - Correct order: [99, 73, 42, 15, 5]

Level 2: Sorting Visualization

- Objective: The player must predict the final sorted array after visualizing a bubble sort operation.
- Puzzle Logic: The game displays a random array and instructs the player to predict what the sorted array will look like after performing a bubble sort.

Example:

- Given array: [3, 6, 2, 5, 1, 4]
- The player must predict the sorted array: [1, 2, 3, 4, 5, 6]

Level 3: Graph Connectivity Puzzle

- Objective: The player is presented with a graph and must determine if all nodes in the graph are connected.
- Puzzle Logic: The game generates a graph with some nodes disconnected. The player is asked if the graph is fully connected. If the player answers correctly, they win the game.

Example:

- Given graph:
- 1 → 2 → 4
- ↓ ↑
- 3 ← 5 (disconnected)
- The player should answer "No" because node 5 is disconnected.

**5. User Interface (UI) Overview:**

The user interface is designed to be simple, intuitive, and visually appealing:

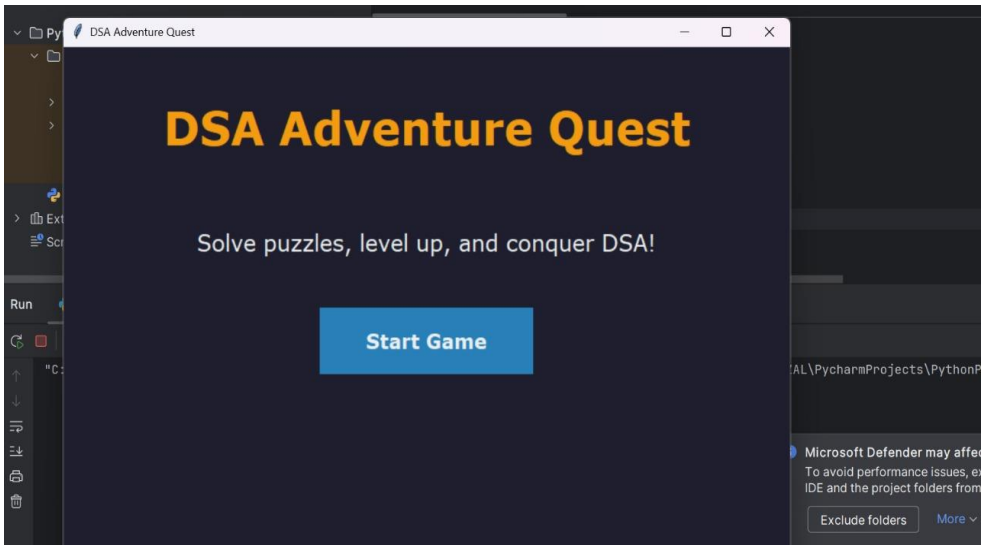
- Main Menu: Displays the game title and a button to start the game.
- Puzzle Screens: For each level, the UI displays instructions, the puzzle itself, an input field for user responses, and a submit button.
- Success and Error Messages: After submitting an answer, the user receives feedback in the form of pop-up messages that inform them if their answer is correct or incorrect.

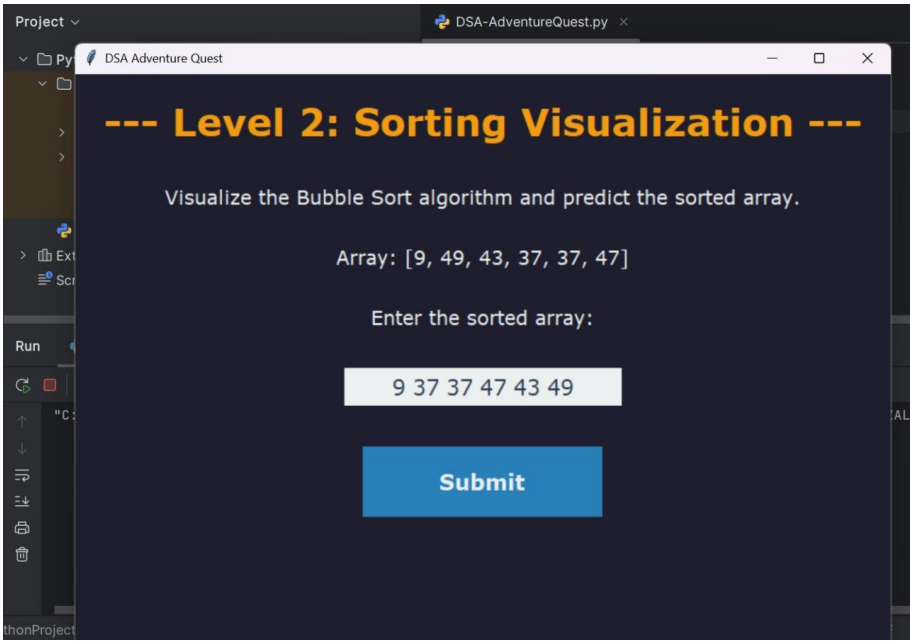
The UI uses a consistent color scheme, with a dark background (#1e1e2e), light text (#ecf0f1), and interactive elements highlighted in blue (#2980b9).

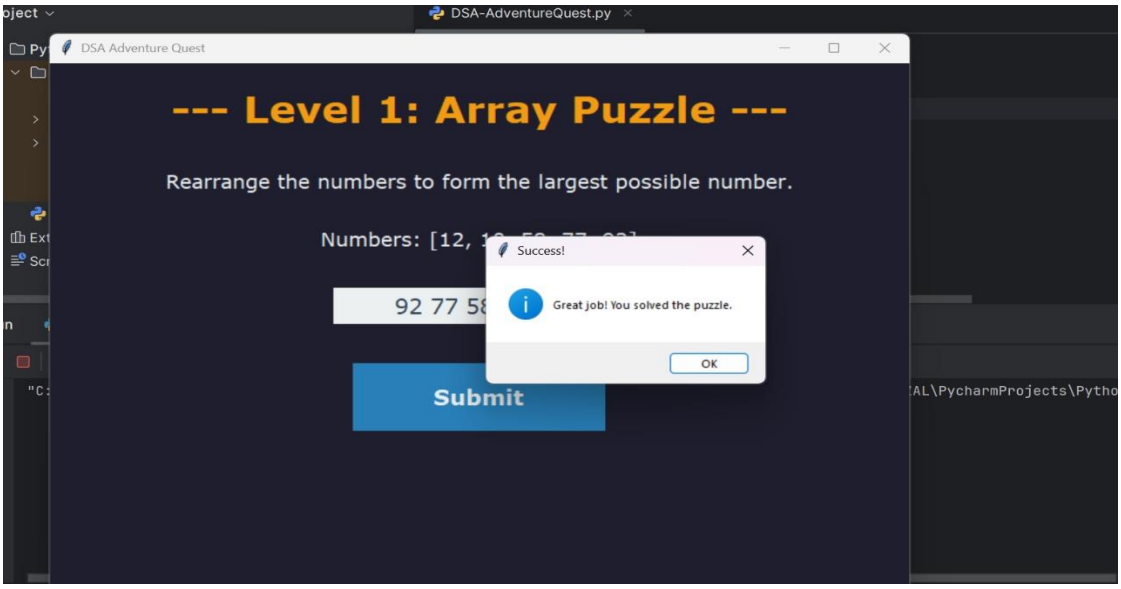
## 6. Detailed Code Breakdown:

1. Game Initialization:
    - The game starts by creating an instance of the DSAAdventureQuest class, which initializes the game window, sets up the title, and presents the main menu.
2. Level 1 (Array Puzzle):
    - Random numbers are generated and displayed on the screen. The player must rearrange them to form the largest number. The input is validated, and if correct, the game proceeds to the next level.
3. Level 2 (Sorting Visualization):
    - A random array is shown, and the player is asked to predict the final sorted array after a bubble sort operation.
4. Level 3 (Graph Connectivity):
    - The player is shown a graph with nodes and edges. They must determine if the graph is fully connected. The game ends after this level.
5. Game Over:
    - Once the player answers the final puzzle, the game shows their score and offers the option to play again or exit.

## 6. Screenshots:




Project
DSA-AdventureQuest.py
DSA Adventure Quest
--- Level 2: Sorting Visualization ---
Visualize the Bubble Sort algorithm and predict the sorted array.
Array: [9, 49, 43, 37, 37, 47]
Enter the sorted array:
9 37 37 47 43 49
Submit
Run


DSA-AdventureQuest.py
DSA Adventure Quest
--- Level 1: Array Puzzle ---
Rearrange the numbers to form the largest possible number.
Numbers: [12,
92 77 5
Submit
Success!
Great job! You solved the puzzle.
OK

## 8. Conclusion:

The DSA Adventure Quest provides a fun and engaging way to learn key concepts of Data Structures and Algorithms through interactive puzzles. Each level introduces a new DSA concept, helping players understand and apply their knowledge in a hands-on way. The use of tkinter makes the game visually appealing and user-friendly, while the random generation of puzzles ensures that no two games are the same.

This project serves as an excellent educational tool for those looking to enhance their DSA skills while enjoying a challenging game.