General Report on Password Generation Using Grid-Based Keyboard Modeling

1. Overview of the Problem

The task involves generating an 8-character password based on the user's initial character selection and subsequent suggestions derived from keyboard character positions. The process utilizes the Manhattan distance on a 2D keyboard grid to calculate valid character moves.

2. Manhattan Distance for Character Positioning

Manhattan distance is a grid-based metric that calculates the distance between two points as the sum of the absolute differences of their row and column indices. This approach is suitable for the keyboard's 2D layout.

3. Data Structures for Efficient Implementation

To efficiently calculate and retrieve valid character moves:

- **Map Data Structure:** A persistent map is used to store the (row, column) positions of each character on the keyboard.
- **Precomputed Moves Map:** A `Map<Character, List<Character>>` maps each key to its valid move list. This avoids recalculating distances for every character.

4. Password Generation Algorithm

The password generation process is implemented as follows:

1. Start with the user-selected character.
2. Retrieve the list of valid moves for the current character.
3. Randomly select the next character from this list.
4. Repeat steps 2 and 3 until the password reaches 8 characters.

5. Example Valid Moves

- For 'a': z, q, s, w
- For 'f': r, t, g, v
- For 'h': y, u, j, b, n
- For '8': 5, 6, 7, 9, i, o

Conclusion

The proposed approach effectively combines grid-based modeling with efficient data structures to generate secure and dynamic passwords. By precomputing valid moves, the algorithm ensures quick and reliable password generation while maintaining flexibility for various keyboard layouts.

Barış PALA

20232810024