ECON381 Fall 2024
Homework Assignment 3

1. The user will select the first character to type. You will then suggest the remaining.
2. For every character selected, you will find a short list of characters that are at a minimum of two but maximum of three moves away from the selected character. For example from the character g, d is 2 moves, s is 3 moves away.
3. You will then select a random character from this short list. This will be your next character.
4. Stop selecting characters when you have a password of size 8.

3.1

The Manhattan distance can be used because the positions of characters on the keyboard are modeled on a 2D grid. Manhattan distance is a solution commonly used in grid-based architectures.

3.2

Yes, a `Map` data structure can be used. This structure is necessary to quickly get the position (row, column) of each character on the keyboard as a key-value pair. It must be stored persistently because we need to recalculate the distances for each new starting character or subsequent characters.

3.3

A `Map<Character, List<Character>>` can be used to map each key to its possible move list. This allows us to quickly retrieve valid moves for each character, so we don't need to recalculate distances repeatedly.

3.4

```
public static String generatePasswordWithPrecomputedMoves(Map<Character,
List<Character>> moves, char start) {
    StringBuilder password = new StringBuilder();
    password.append(start);
    Random random = new Random();
    char current = start;

    while (password.length() < 8) {
        List<Character> validMoves = moves.get(current);
        if (validMoves == null || validMoves.isEmpty()) {
            throw new IllegalStateException("No valid moves for character:
" + current);
        }
        current = validMoves.get(random.nextInt(validMoves.size()));
        password.append(current);
    }

    return password.toString();
}
```

3.5

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

Barış PALA
20232810024