ECON381 FALL 2024

Homework Assingment 3

Question 1: Which distance metric is usable for distances between keys?

Answer: Manhattan Distance is available as a distance measurement in a virtual keyboard layout. This metric refers to the total number of Extreme and vertical movements to reach from one key to the next. For example:

If the 'a' key is at the (0,0) coordinate and the 's' key is at the (0,1) coordinate, the distance

$= |0 - 0| + |1 - 0| = 1.$

Question 2: Would you need a particular data structure to represent the keyboard layout? Would this structure be needed permanent or once we calculate the distances between keys, could it be replaced by another structure?

Answer: To represent the keyboard layout and efficiently calculate distances between keys, you would initially need a data structure. The choice and permanence of this structure depend on the requirements and the specific tasks.

Once you calculate distances, the keyboard structure may not be needed permanently. You could replace it with a precomputed distance matrix or a more efficient structure to speed up further computations.

Question 3: Suppose we decided to map each key to a list of valid moves (ie. other keys with 2 to 3 distance). What kind of Java data structure be the best suited fort his?

Answer: A hash table can be used to assign a character a list of other valid characters.

Each key is stored as a key-value pair containing a list of valid moves.

Question 4: Write pseudocode (or Java code) for creating an 8 character password using the data structure you suggested.

Answer:

Java Code example:

```
import java.util.*;


public class PasswordGenerator {

   public static void main(String[] args) {
```

```
        Map<Character, List<Character>> keyMap = new HashMap<>();
        keyMap.put('a', Arrays.asList('q', 's', 'z'));
        keyMap.put('s', Arrays.asList('a', 'w', 'd', 'x'));
        keyMap.put('d', Arrays.asList('s', 'e', 'f', 'c'));

        String password = generatePassword(keyMap, 8);
        System.out.println("Generated Password: " + password);
    }

    public static String generatePassword(Map<Character, List<Character>> keyMap, int length) {
        Random random = new Random();
        List<Character> keys = new ArrayList<>(keyMap.keySet());
        char currentChar = keys.get(random.nextInt(keys.size())); // Rastgele başlangıç
        StringBuilder password = new StringBuilder();
        password.append(currentChar);

        while (password.length() < length) {
            List<Character> validMoves = keyMap.get(currentChar);
            currentChar = validMoves.get(random.nextInt(validMoves.size()));
            password.append(currentChar);
        }

        return password.toString();
    }
}
```

Question 5: Compute the list of valid moves for for the following keys: a, f, h, 8, 0, and p.

Answer:

Given keys: a, f, h, 8, 0, p.

For a: q, s, z.

For f: d, g, v.

For h: g, j, n.

For 8: 7, 9, i, o.

For 0: 9, p.

For p: o, l, 0.

These lists can be stored with a hash table.