ECON 381 HOMEWORK 3

KÜBRA KILIÇ 20232810010

**1.** Manhattan distance, sometimes called L1 distance, is a convenient distance measure. This measurement sums the absolute coordinate differences between two locations. The distance between these locations is determined. Determines the amount of horizontal and vertical keystrokes required to cycle through keys on a Keyboard. Single movements do not include diagonal movements.

**2.** The best data structure for specifying a keyboard layout is usually a matrix. Matrices are 2-dimensional arrays. There is a key for each element in the array. This aims to use array indices as coordinates to calculate Manhattan distances.

This structure needs to be calculated and saved at least once. Theoretically you can calculate distances on the fly, but this is ineffective. By storing the keyboard layout, creating passwords is accelerated. The 2D array is required for initial distance calculations, but once the distances are calculated we can transform the 2D array representation into a more efficient search structure.

**3.** The most suitable Java data structure is HashMap<Character, List<Character>>.

Key: The start key is represented by character.

Value: Characters two or three moves away from the key are stored in the <Character> List.

For any given key, this structure allows a fast search of valid moves.

**4.** Java code file uploaded separately

**5.** Depending on the given keyboard layout, the following gestures (2 or 3 distances) are acceptable for designated keys:

**For a:** e, x, c, v, y, w

**For f:** w, r, a, d, g, z, x, c, t, e, s

**For h:** r, y, u, f, g, c, v, b, n, d, t, j

**For 8:** o, q, w, e, i, 0, 4, 5, 6, 9,

**For 0:** p, u, i, o, 6, 7, 8, 9

**For p:** i, o, 0, 7, 8, 9