Keyboard Layout Optimization

Objective

- Visualize how keyboard layout affects typing effort.
- Use simulated annealing to optimize a keyboard layout for a given text.

Problem

You are given a baseline QWERTY layout with fixed key coordinates. A layout is a mapping from characters to coordinate positions. Define the cost of a layout for an input text as the total Euclidean distance traveled across consecutive characters (ignore case; treat unknown characters as a space). Your task is to:

- Implement a simulated annealing optimizer that explores layouts by swapping key assignments while keeping coordinates fixed.
- Minimize the path-length cost for a provided text (single string or text file).
- Produce simple visualizations:
  - A scatter plot of keys at their coordinates with labels for the final layout.
  - A line plot of best cost vs. iteration.

Note: The original inspiration for this was from https://www.patrick-wied.at/projects/heatmap-keyboard/ This page shows a heatmap that shows how often a key is pressed.

This is not how the cost would be computed on a real keyboard: the actual distance traveled is not from one letter to the next. It should be computed in terms of how far each key is from the "home" position that your fingers are placed on. So for example, the letter "f" would have 0 travel distance since you are supposed to always have one finger on it, while "e" would require moving from "d" to "e" and back. This is harder to model since we now need to decide how the home row is set up, so we omit this from the problem statement. The cost being computed here is more like "hunt and peck" where we assume you are typing with just one finger.

This is a simplified version of the keyboard optimization problem. Note that for the purpose of the course, this given problem statement is what matters. Adding more features will not necessarily result in more marks.

Requirements

- Start from the provided QWERTY coordinates (same positions, different assignments allowed).

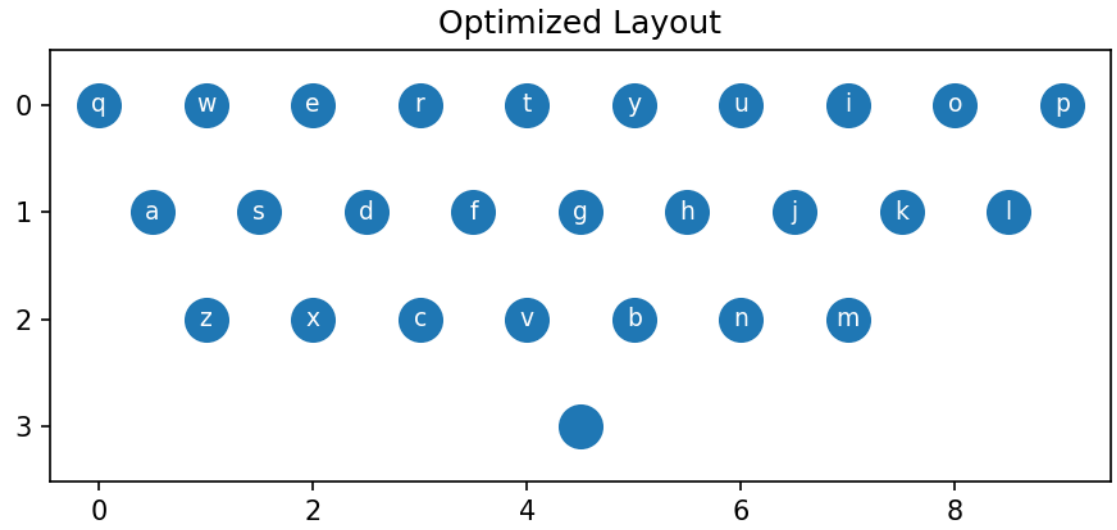
- Optimization move: swap the positions of two characters.
- Input: a string or a path to a text file.
- Output: final layout (mapping), final cost, and plots described below.
- Keep your code clear and concise; comment key choices and parameters.

Given

- Script skeleton: `kbd_optim_base.py` - this contains only a few function names that may be useful. You need to fill out the content. If you choose to redo with different functions then document this clearly.
  - Functions to create a basic layout and plot the result are given
- Sample text: `assignments/kbd_optim/sample_text.txt`

Format for keyboard layout

The plotting function given will generate a plot that looks like the following, but should contain the keys in the updated positions that you have found for optimizing the travel distance.



Submission

You need to submit the following, in a single file called rollno_A3.zip (use your roll number).

- Code that runs end-to-end from the command line on the starter data. File should be named as `kbd_optim.py`
- A PDF file (`readme.pdf`) that briefly explains your code. Not more than 2 pages. This should include:
  - at least two figures:
    * plot of the loss function over time that shows how the optimization happened
    * final layout of the keyboard
  - JSON format
  - brief discussion of how the result changes as the temperature and number of iterations are varied. You may add more figures to explain this if needed, but only what is important: the report should not become too long.
  - Exact instructions on how to run your code with different settings like temperature or number of iterations.