CSE 2010, Term Project, Spring 2023

Due Tue Apr 18 at the start of your lab section; Submit Server: class = cse2010, assignment = termProjectSxInitial

Due Tue Apr 25 at the start of your lab section; Submit Server: class = cse2010, assignment = termProjectSxFinal

x is 14 or 23—your section number (or c for c language)

When you search using a search engine, the browser and/or the search engine provides suggestions to reduce typing. How would you design a system that can help you and other users type quickly and accurately?

The challenge of the term project is to design and build QuerySidekick, a system that can guess queries based on partial queries. For each query, given a character at a time, your system will provide 5 guesses. If one of the guesses is correct, the rest of the letters are skipped (yeah!). You and other users desire accuracy, speed and low memory usage, which can be measured by:

- accuracy (average % of letters skipped for each query)
- average time used to generate 5 guesses for a query
- number of bytes used in the memory
- $score = accuracy^2/\sqrt{time * memory}$

Program Files: Among your program files, you will provide QuerySidekick.java that supports at least:

- procOldQueries(...) // old queries
- guess(...) // 5 guesses on new queries
- feedback(...) // feedback for QuerySidekick

We will provide EvalQuerySidekick.java (not to be modified) to help you evaluate your system. EvalQuerySidekick.java and a template of QuerySidekick.java with details of the above functions are on the course website. The preprocessing of old queries should not exceed 10 minutes.

You may use any program segment/file from the textbook and standard Java libraries without additional installation. However, you may *NOT* use program segments/files from other sources. You may borrow ideas from other sources, but you need to clearly cite them at the top of your program files and implement them on your own.

Input: Input is from the command-line arguments for EvalQuerySidekick.java in this order:

- filename for old queries to simulate old user input, one query per line
- filename for new queries to simulate new user input, one query per line

Sample input files are on the course website.

You may have additional "hardcoded" (not from command line or prompt) input text/word files (such as the provided words.txt) for QuerySidekick.java and/or other program files.

Output: Measured performance by EvalQuerySidekick.java goes to the standand output (screen).

Presentation: Project presentations (about 10 minutes each) will be during the lab on Apr 25 (Tue). Create a presentation file following this outline:

1. Title, Group Name, and Members
2. Goal and Motivation
3. Initial Approach/Submission
   (a) Algorithms and supporting data structures
   (b) Additional input data and reasons
   (c) Ideas devised within the group
   (d) Ideas discussed in the course/book, cite them
   (e) Ideas borrowed from other sources, cite them
4. Final Approach/Submission
   (a) Changes in algorithms and supporting data structures
   (b) Changes in additional input data and reasons
   (c) Ideas devised within the group
   (d) Ideas discussed in the course/book, cite them
   (e) Ideas borrowed from other sources, cite them
5. Evaluation: Accuracy, time, memory, and score: initial vs. final approach/submission for each data set
6. Analysis
   (a) Why more/less accurate?
   (b) Why faster/slower (including time complexity—n queries, each query has m characters)?
   (c) Why more/less memory?
   (d) Possible further improvements

Evaluation:

- Initial submission (20%) – including performance on accuracy and score on code01.fit.edu
- Final submission (40%) – including performance on accuracy and score on code01.fit.edu
- Presentation (20%, oral and written)
- Teammate evaluation (20%)
- Extra Credit (a max of 10 points): Final Submission has an accuracy above 80, using less than 1×10^{-4} (0.0001) seconds and 7×10^7 bytes (~70 MB), as measured by EvalQuerySidekick.java on multiple data sets on code01.fit.edu. Average accuracy of at least 81 earns one extra point, at least 82 earns two extra points, ... Late submission is not applicable.

Submission: Initial submision has QuerySidekick.java, other program files, and additional data files (if any).

Final submission has QuerySidekick.java, other program files, additional data files (if any), and presentation (preferably in PDF).

Note the late penalty on the syllabus if you submit after the due date and time as specified at the top of the assignment.