Carnegie Mellon University
15-826 – Multimedia Databases and Data Mining
Spring 2009, C. Faloutsos
Assignment 2, Due Date: **March 17**, in class

Reminders

- Due date: March 17, 3:00pm, hard copy in class and soft copy e-mailed to the TA (*bziebart+826 at cs*) in a single e-mail (along with your source code).
- Please turn in a **typed** report.
- All homeworks including this one are to be done INDVIDUALLY.
- Expected effort for this homework (approximate times):
  - Q1: 3 hours
    - 2 hours to write and run a program or recursively decompose and solve by hand
    - 1 hour to write answers for the questions
  - Q2: 4 hours
    - 1 hour to download and parse the dataset
    - 2 hours to write and debug your algorithm
    - 1 hour to run your algorithm and answer questions
  - Q3: 4 hours
    - 1 hour to download and make the package
    - 2 hours to write and debug your algorithm
    - 1 hours to run your algorithm and answer questions
  - Q4: 4 hours
    - 1 hour to download existing code
    - 2 hour to write new penalties for typos
    - 1 hour run algorithms and answer questions
  - Q5: 5 hours
    - 2 hours to download and load text
    - 2 hours to analyze data
    - 1 hour to answer questions

Q1 – Hilbert Curve and Z-Ordering [20pts]

Problem Description: We will now analyze properties of the Hilbert Curve and Z-Orderings. Consider a KxK query, Q, in a larger NxN grid and some ordering over the entire NxN grid. We will consider queries inside the grid with no wrap-around. Each cell in the query Q has a value in the ordering. We are interested in the number of different runs in Q, **runs**(Q). See slide 110 in Lecture #5: Multi-key and Spatial Access Methods II.

1. For a Hilbert Curve and K=2 and N=4, what is the worst case, $\max_Q \mathbf{runs}(Q)$
2. For a Hilbert Curve and K=2 and N=4, what is average case, $\frac{1}{|Q|}\sum_Q \mathbf{runs}(Q)$?
3. For a Hilbert Curve and K=2 and N=8, what is average case, $\frac{1}{|Q|}\sum_Q \mathbf{runs}(Q)$?
4. For a Hilbert Curve and K=3 and N=8, what is average case, $\frac{1}{|Q|}\sum_Q \mathbf{runs}(Q)$?
5. For a Z-Ordering and K=2 and N=4, what is the worst case, $\max_Q \mathbf{runs}(Q)$?
6. For a Z-Ordering and K=2 and N=4, what is average case, $\frac{1}{|Q|}\sum_Q \mathbf{runs}(Q)$?
7. For a Z-Ordering and K=2 and N=8, what is average case, $\frac{1}{|Q|}\sum_Q \mathbf{runs}(Q)$?
8. For a Z-Ordering and K=3 and N=8, what is average case, $\frac{1}{|Q|}\sum_Q \mathbf{runs}(Q)$?

Q2 – Fractal Dimension (Measuring) [20pts]

Problem Description: We will implement an algorithm for calculating the correlation integral of a cloud of points and use it on a few 5-dimensional datasets.

1. Implement the naïve $O(N^2)$ algorithm for computing the correlation integral. Your algorithm should include self-pairs and mirror pairs. Please submit your code.
2. Plot and submit the correlation integral for mystery dataset #1 (`http://www.cs.cmu.edu/~bziebart/15826-S09/hw2/mystery1.dat`).
3. Plot and submit the correlation integral for mystery dataset #2 (`http://www.cs.cmu.edu/~bziebart/15826-S09/hw2/mystery2.dat`).
4. Plot and submit the correlation integral for mystery dataset #3 (`http://www.cs.cmu.edu/~bziebart/15826-S09/hw2/mystery3.dat`).

Note: If there are too many points to plot, sample logarithmically.

Q3 – Fractal Dimension (Generating) [20pts]

Please generate points that obey the correlation integral of Figure 1. You may use the fractal-dimension code available from `http://www.cs.cmu.edu/~christos/SRC/fdnq.tar` (feel free to modify), other code available from the web, or write your own code.

What to hand in:

1. Briefly describe the mechanism used to generate your dataset.
2. Using your code from Question 2, or `fdnq.tar` mentioned above, measure and plot the correlation integral. Note: Your plot does not need to have perfectly sharp corners at break points.

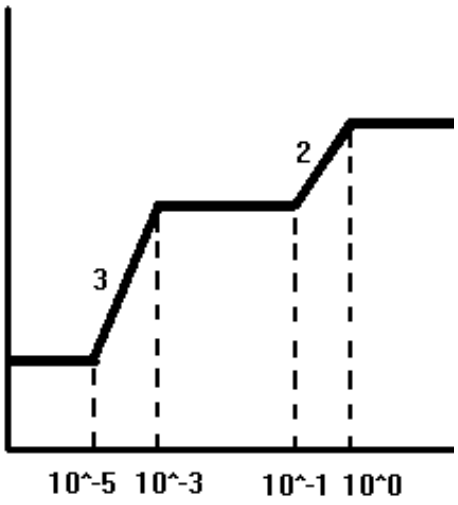

Figure 1: Correlation integral.

Q4 – String Edit Distance [20pts]

Problem Description: The existing string editing distance code `http://www.cs.cmu.edu/~bziebart/15826.S09/hw2/stredit.pl` penalizes insertion, deletions and substitutions by 1 unit.

1. Modify this code to weakly penalize common typos. We will use the keyboard character distance on a standard QWERTY keyboard to determine typo "cost". For example, *substitution-cost(q,a) = substitution-cost(a,z) = substitution-cost(a,s) = .05.* All other costs do not change (cost = 1.0 for insertion, substitution, deletion). The full set of pairs of letters with cost .05 is available at `http://www.cs.cmu.edu/~bziebart/15826-S09/hw2/pairs.txt`. Please HAND IN your code.

2. Consider the set of typo words available at `http://www.cs.cmu.edu/~bziebart/15826-S09/hw2/typos.txt`, and the set of dictionary words available at `http://www.cs.cmu.edu/~bziebart/15826-S09/hw2/dictionary.txt`. For each typo, use your typo distance function to obtain the closest dictionary word (break ties alphabetically – with 'a' preferable to 'z'). Report the the typo word, the dictionary word and the cost between typo word and dictionary word.

Q5 – Power Laws in Text [20pts]

The inter-arrival time measures the number of words between two consecutive occurrences of a query word. We will denote this with random variable W.

For example, in the sentence *"The quick brown fox jumps over the lazy dog,"* there is a single inter-arrival occurance, $W = 6$ for the word *the*.

Zipf claims that the inter-arrival times of common words follow a power law distribution. We will now verify this claim.

Download the follow electronic books from the Project Guttenberg website. You will need to decompress them and remove the header text (up to and including *START OF THE PROJECT GUTENBERG EBOOK...*) in the uncompressed file and the trailing text about Project Gutenberg (including and after *END OF THE PROJECT GUTENBERG EBOOK*).

- The Count of Montre Cristo – `http://www.gutenberg.org/etext/1184`
- The Art of War – `http://www.gutenberg.org/etext/132`
- The Adventures of Sherlock Holmes – `http://www.gutenberg.org/etext/1661`

1. Plot the inter-arrival time distribution, W, of the word "the" for each of the texts in log-log scale (inter-arrival time on the x-axis, number of occurrences on the y-axis).
2. Fit a line to the data for each text and report the slope of this line.

Hints:

- Text files can be converted into having one word per line using the `tr -cr` command in linux/unix/cygwin (e.g., `tr -cs 'a-zA-Z' '[\n*]' < input-file.txt > words-lines.txt`).
- Words can be converted to lower case using: `tr A-Z a-z < input-file.txt > lower-case.txt`.
- Line numbers can be added to a file using: `cat -n < input.txt > numbered-output.txt`.