CSCI 393, Algorithm Design and Analysis, Spring 2020

HW 04 – Closest and Farthest Pair of Words

Assignent

Implement a Brute Force algorithm to find the closest and farthest pair of 5-letter words contained in the text file `words.txt` (a link to the text file is on the class website). You can implement the algorithm in Java, Python, or C++.

What to Turn In

Submit the source code file for your program through Blackboard.

Discussion and Details

In class we discussed a Brute Force algorithm to find the closest pair of points from a set of points in the Euclidean Plane. In that example, for the distance between two points $p_1 = (x_1, y_1)$ and $p_2 = (x_2, y_2)$ we used the formula

$$dist(p_1, p_2) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

For this assignment you are to write a program that implements that same algorithm, only with two changes. Instead of looking at a set of points, you will be looking at a set of 5-letter words. Also, your program should find both the closest pair of words *and the farthest* pair of words. In order to find the two words that are the "closest" and the two that are the "farthest", we need to define the distance between to words.

We'll begin by looking at how to take a measurement of a single 5-letter word consisting of the characters $(c_4, c_3, c_2, c_1, c_0)$. For each character c_i, let a_i be the character's ASCII code (note $0 \leq a_i \leq 255$). Also let r be a radix, where $r > 255$. Then the measurement of word w is given by

$$measure(w) = a_4 r^4 + a_3 r^3 + a_2 r^2 + a_1 r^1 + a_0 r^0$$

and the distance between two words, w_1 and w_2 is given by

$$dist(w_1, w_2) = |measure(w_1) - measure(w_2)|$$

Example

Let's compute the distance between the words `cable` and `scoff`, with $r = 256$.

`cable`
The characters are: $c_4 = c$, $c_3 = a$, $c_2 = b$, $c_1 = l$, $c_0 = e$
The ASCII codes for these letters are: $a_4 = 99$, $a_3 = 97$, $a_2 = 98$, $a_1 = 108$, $a_0 = 101$

$$\begin{aligned} measure(\texttt{cable}) &= a_4 r^4 + a_3 r^3 + a_3 r^2 + a_1 r^1 + a_0 r^0 \\ &= (99)(256^4) + (97)(256^3) + (98)(256^2) + (108)(256^1) + (101)(256^0) \\ &= (99)(4294967296) + (97)(16777216) + (98)(65536) + (108)(256) + (101)(1) \\ &= 426835602533 \end{aligned}$$

`scoff`
The characters are: $c_4 = s$, $c_3 = c$, $c_2 = o$, $c_1 = f$, $c_0 = f$
The ASCII codes for these letters are: $a_4 = 115$, $a_3 = 99$, $a_2 = 111$, $a_1 = 102$, $a_0 = 102$

$$\begin{aligned} measure(\texttt{scoff}) &= a_4 r^4 + a_3 r^3 + a_3 r^2 + a_1 r^1 + a_0 r^0 \\ &= (115)(256^4) + (99)(256^3) + (111)(256^2) + (102)(256^1) + (102)(256^0) \\ &= (115)(4294967296) + (99)(16777216) + (111)(65536) + (102)(256) + (102)(1) \\ &= 495589484134 \end{aligned}$$

Then the distance between `cable` and `scoff` is

$$\begin{aligned} dist(\texttt{cable}, \texttt{scoff}) &= |measure(\texttt{cable}) - measure(\texttt{scoff})| \\ &= |426835602533 - 495589484134| \\ &= |-68753881601| \\ &= 68753881601 \end{aligned}$$

Program Input

Your program should not prompt for or read any user input. The program should open the text file `words.txt` and read in the words. The text file should be in the default file location. This is the only input to your program.

Required Output

Your program should print the following:

1. The radix used for computing the measurement of each word.
2. The smallest distance between any two words.
3. Two words that are the closest.
4. The largest distance between any two words.
5. Two words that are the farthest.