CSE2010 Term Project Spring 2019 Hangman Player

Team: " `Fantastic for(int i=0; i<4; i++)`"
Sarah Arends, Ale Escobar, Michael Hon, Josias Moukpe

I. Goal and Motivation

THE project serves to simulate a game of Hangman. Our program will interface with an evaluation program that provides a list of unknown words and returns feedback on guessed letters. The "Hangman Player" program will initialize data structures from a dictionary file, use known information to guess a subsequent letter, and perform additional operations in response to feedback from the evaluation program. This project tests our ability to efficiently match patterns in strings.

II. Initial Approach

A. Algorithms and Supporting Data Structures

B. Additional Input: Headers

C. Ideas Devised by Group

D. Ideas Discussed in Course and Material

E. Ideas from Other Sources

III. Final Approach

A. Changes in Algorithms and Supporting Data Structures

Rather than storing each word as an ordered path of characters in a trie structure, the information on each word is encoded in the following structure.

```
typedef struct {

    byte_t freqs_indices[ALPHABET_SIZE];
    freq_t* letter_freqs;
    bool is_cand;

} Word_t;
```

Listing 1. Word Struct

The array `freqs_indices` contains the frequency of each letter in the word.

The boolean `is_cand` indicates whether the given word is still a candidate, meaning it matches the known information about the hidden word thus far. This is **bolddd**. I want to make an inline equation that says $f(x) = \lambda^2$.

B. Additional Input: Headers

C. Ideas Devised by Group

D. Ideas Discussed in Course and Material

E. Ideas from Other Sources

IV. Evaluation

A. Accuracy, Time, and Memory usage

B. Improvements from Initial to Final Submission

V. Analysis

A. Improvement in Accuracy

B. Improvement in Time Complexity

C. Improvements in Memory Usage

D. Possible Further Improvements