All functions

Functions

std::string	**fr_dict::find_word** (const std::string &target) Performs search to find the target word in Complexity $O(N)$ because it needs to hash the target string first, where N is the size of the string.
string	**fr_dict::search_definition** (int desired_row_number) Looks for the stored definition of the word in the dictionary. It has complexity $O(N)$ as it gets the entire line until it finds the number of line.
void	**fr_dict::load_dictionary** () Loads the words into the hash_dictionary in $O(N)$ time since it goes line by line.
void	**fr_dict::print_dictionary** () Prints the hash dictionary in $O(N)$.
bool	**fr_dict::first_run** () Checks if it is the first run. $O(1)$.
void	**fr_dict::process_xml** (unsigned int entries) Filters the selecteed amount of entries in the xml file. $O(N^2)$.
	HashTable::HashTable () Construct a new Hash Table object, allocate item positions. $O(N)$.
int	**HashTable::hash_function** (const std::string &str) simple hash function that increases the value on each loop according to the integer value of the char. $O(N)$ where N is the str length.
void	**HashTable::insert** (const std::string &str) Linear probing to insert value. Worst case is $O(N)$ for the last value if the keys happen to be poisoned (they are generated sequentially).
std::string	**HashTable::search** (const std::string &str) Searches in $O(N)$ time where N is the length of the str to hash.
void	**HashTable::print_hash_table** () Prints every location with a value in it. $O(N)$.
bool	**Preprocess::not_first_run** () check if the preprocessing has been ran before. Complexity $O(1)$.
void	**Preprocess::filter_xml_data** (unsigned int entries) Filters data using the predefined regex patterns. Dumps definitions and words to separate files. Complexity $O(n^2)$.

Detailed Description

This is the functions group with its asymptotic analysis

Function Documentation

◆ filter_xml_data()

void Preprocess::filter_xml_data (unsigned int entries) inline

Filters data using the predefined regex patterns. Dumps definitions and words to separate files. Complexity $O(n^2)$.

Parameters

entries is the number of dictionary entries to acquire

◆ find_word()

std::string fr_dict::find_word (const std::string & target) inline

Performs search to find the target word in Complexity $O(N)$ because it needs to hash the target string first, where N is the size of the string.

Parameters

target Word to look for

Returns

int Index of the target word

See also

HashTable::search();

◆ first_run()

bool fr_dict::first_run ( ) inline

Checks if it is the first run. $O(1)$.

Returns

true if it is the first run

false if it isnt the first run

See also

Preprocess::not_first_run();

◆ hash_function()

int HashTable::hash_function (const std::string & str) inline

simple hash function that increases the value on each loop according to the integer value of the char. $O(N)$ where N is the str length.

Parameters

str String to hash

Returns

int Calculated key value of the string

◆ HashTable()

HashTable::HashTable ( ) inline

Construct a new Hash Table object, allocate item positions. $O(N)$.

◆ insert()

void HashTable::insert (const std::string & str) inline

Linear probing to insert value. Worst case is $O(N)$ for the last value if the keys happen to be poisoned (they are generated sequentially).

Parameters

str

◆ load_dictionary()

void fr_dict::load_dictionary ( ) inline

Loads the words into the hash_dictionary in $O(N)$ time since it goes line by line.

◆ not_first_run()

bool Preprocess::not_first_run ( ) inline

check if the preprocessing has been ran before. Complexity $O(1)$.

Returns

true it is NOT the first, files must NOT be generated

false it IS the first run and the files must be generated

◆ print_dictionary()

void fr_dict::print_dictionary ( ) inline

Prints the hash dictionary in $O(N)$.

See also

HashTable::print_hash_table();

◆ print_hash_table()

void HashTable::print_hash_table ( ) inline

Prints every location with a value in it. $O(N)$.

◆ process_xml()

void fr_dict::process_xml (unsigned int entries) inline

Filters the selecteed amount of entries in the xml file. $O(N^2)$.

Parameters

entries words to be processed

See also

Preprocess::filter_xml_data();

◆ search()

std::string HashTable::search (const std::string & str) inline

Searches in $O(N)$ time where N is the length of the str to hash.

Parameters

str String to hash and look for

Returns

std::string

◆ search_definition()

string fr_dict::search_definition (int desired_row_number) inline

Looks for the stored definition of the word in the dictionary. It has complexity $O(N)$ as it gets the entire line until it finds the number of line.

Parameters

desired_row_number

Returns

string "Not found" if there is no definition, "Could not open file" if the file has been corrupted or the definition if it is found.