Introduction of rucgraph

Folder: build_in_progress

This folder contains informal codes.

Folder: data_structures

This folder contains some special data structures.

PairingHeapYS.h

This file contains a pairing heap.

PairingHeapYS_with_offset.h

This file contains an augmented pairing heap. In this heap, there is an offset value for every inside node. Using these values, we can change the key values of all inside nodes in O(1) time!

Union_Find.h

This file contains the Union Find data structure.

Folder: graph_hash_of_mixed_weighted

Folder: graph_hash_of_vectors_unweighted

graph_hash_of_vectors_unweighted.h

This is an adjacency list build using a hash of vectors:

```
/* define graph: a hash of vectors */
typedef std::unordered_map<int, std::vector<int>> graph_hash_of_vectors_unweighted;
```

This adjacency list does not contain vertex or edge weights.

Folder: graph_hash_of_vectors_weighted

graph_hash_of_vectors_weighted.h

This is an adjacency list build using a hash of vectors:

```
/* define graph: a hash of vectors */
class graph_hash_of_vectors_weighted_vertex_content {
public:
    double vertex_weight; // weight of this vertex
    std::vector<pair<int, double>> adj_vertices; // adjacenct vertices and weights of edges; ordered from small to large
};
typedef std::unordered_map<int, graph_hash_of_vectors_weighted_vertex_content> graph_hash_of_vectors_weighted;
/*this is an undirected, static graph*/
```

This adjacency list contains vertex or edge weights.

Folder: graph_v_of_v_idealID

Folder: text_mining

binary_save_read_vector

This is to save and read vectors in binary format. Notably, the elements in vectors should have fixed sizes.

binary_save_read_vector_of_vectors.h

This is similar to the above file, for saving and reading vectors of vectors.

convert_number_to_array_of_binary.h

This is to concert a number to an array of binary values, e.g., from 3 to 11.

latitude_and_longitude_distance.h

This is to compute the distance between two points using latitude_and_longitude.

list_all_files_in_a_directory.h

This is to list all file names in a path.

parse_string.h

This is to parse a string based on a delimiter.

parse_substring_between_pairs_of_delimiters.h

This is get substrings between a pair of different delimiters.

parse_substring_between_two_unique_delimiters.h

This is to get the substring between two_unique_delimiters.

print_items.h

This is used to print items.

read_csv.h

This is to read a csv file into a vector of vectors of strings.

read_file_line_by_line.h

This is to print a file line by line.

read_file_total_line_number.h

This is to print the total line number of a file.

replace_chars_in_string.h

This function replace all chars "from" in a string to "to".

string_contains_number.h

This is to check whether a string contains a number char.

string_is_number.h

This is to check whether a string is a number.

StringCompare_caseInSensitive.h

This is an insensitive comparison of two strings, e.g., A==a.

utc_time_to_local_time.h

This is to convert utc_time_to_local_time.

Folder: tool_functions

This folder contains some tool functions.

Combinations_Permutations.h

This file contains codes to enumerate every possible permutation of a set of elements.

Current_Memory_Consumption_of_This_Process.h

This file is to check how many RAM has been allocated by the OS to the current process.

ThreadPool.h

This is a widely adopted ThreadPool.h implementation.