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

This is an adjacency list build using 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

This is an adjacency list build using 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 adjacency list contains vertex or edge weights.

Folder: graph_v_of_v_idealID

Folder: text_mining

Folder: tool_functions