Lab 7 – Exercises

Exercise 1.

(i) Extend the class HashSet by functions for set union, intersection and difference.

(ii) Modify the class HashSet by using parameters for the minimum and maximum load factor (currently set to 0.25 and 0.75).

Exercise 2. There are many approaches that aim to improve the performance of hashing. One possibility is to reorganise the hash table during insertions.

(i) Implement the following method, known as *last-come-first-served hashing*. An element to be inserted is always placed in the position given by its hash value. If this position is occupied, the current "resident" is moved to another position using the search procedure defined for HashSet.

(ii) Compare the performance with the performance of the method on the HashSet class.

Exercise 3.

(i) Implement a variant of linear probing without using placeholders. Instead use the alternative *delete* operation handled in the lecture to move another element into the freed space.

(ii) Implement a variant of linear probing, where the size of the hash table is not m, but $m + m'$. In the search procedure in *insert*, *find* and *delete* increment the search index not considering the residue modulo m.

Exercise 4. Modify the *rehash* operation such that each time a new random hash function is used. Use universal hashing.