

Lab Discussion 5

EXERCISE 1.

- (i) Explore the various sorting methods on the class `HASHSET` defined in the lectures.
- (ii) Extend the class `HASHSET` by methods for checking of equality of sets or subsets and operations for set union, intersection and difference.
- (iii) Modify the class `HASHSET` by using parameters for the minimum and maximum load factor (currently set to 0.25 and 0.75).

EXERCISE 2. A *map* M is a partial function that is defined on a set K , i.e. we can write M as a set of *key-value pairs* (k, v) with unique keys $k \in K$.

Memoisation is a programming technique useful for cases, where the same call of a function is used multiple times. In order to avoid costly recomputation it uses a map to store results from previous calls.

- (i) Define a class `HASHMAP` in analogy to `HASHSET` to represent maps. As keys uniquely determine key-value pairs let the hash values only depend on the keys of such pairs.
- (ii) Use your class `HASHMAP` for memoisation on an example of your choice—you could use a naive algorithm for computing Fibonacci numbers, the towers of Hanoi problem, or just the location of elements on a given position in a doubly-linked list.