

Lab Discussion 6

EXERCISE 1.

- (i) Modify the class `HASHSET` such that hashing with chaining is used instead of linear probing.
- (ii) Explore the definition of further set operations (as in Exercise 1(ii) in the previous lab) on this modified implementation of hashsets.

EXERCISE 2. Modify the *rehash* operation such that each time a new random hash function is used. Use universal hashing.

EXERCISE 3 (LINEAR HASHING). Implement a simple variant of *linear hashing* as an alternative to hashing with chaining. Fix a maximum length l of the lists stored in a table entry. Fix a number m and a hash function h with values in $\{0, \dots, m-1\}$. The hash table will have a size $M \geq m$. Then define inductively hash functions h_i with $h_0 = h$ and $h_{j+1}(e) = h_j(e) + m \cdot 2^j$, so h_j takes $2^j \cdot m$ different hash values.

The hash value of an element e is $h_j(e)$, where j is maximal with $h_j(e) \leq M-1$.

- (i) Show that whenever an element e is given, it suffices to compute at most two hash values.
- (ii) Implement a *rehash* operation that comes into action, when the maximum list length l will be exceeded by insertion of a new element e . In this case use h_{j+1} to distribute the elements of the list into two sublists and increase M accordingly. If necessary, split other lists as well according to the new value M .
- (iii) Show that if it is permitted to use also h_{j-1} instead of h_j to determine the hash value of an element e , then rehashing can be done in a lazy way by splitting only one list.