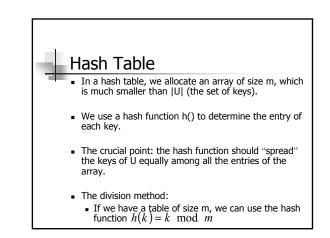
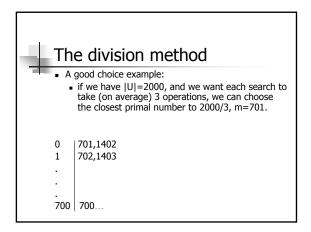
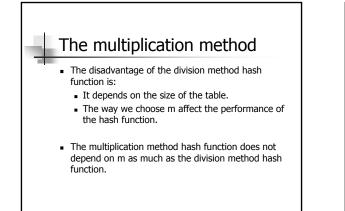
Tirgul 9

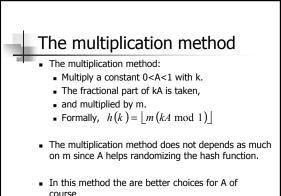
Hash Tables (continued) Reminder Examples

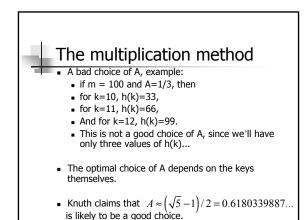


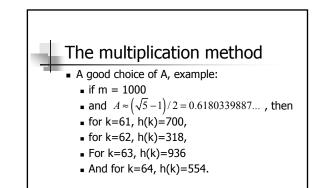
How to choose hash functions The crucial point: the hash function should "spread" the keys of U equally among all the entries of the array. Unfortunately, since we don't know in advance the keys that we'll get from U, this can be done only approximately. Remark: the hash functions usually assume that the keys are numbers. We'll discuss next class what to do if the keys are not numbers.

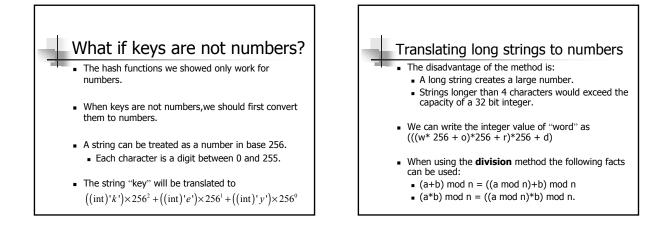


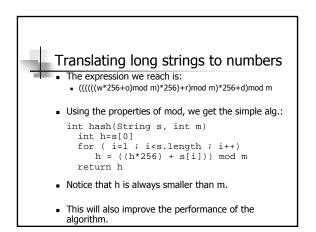


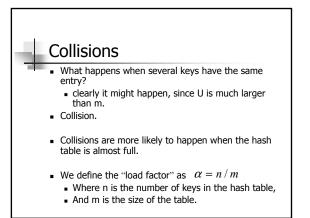








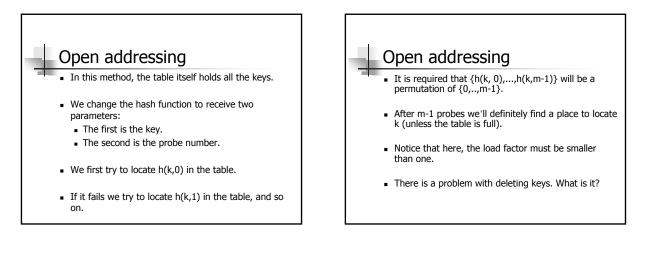


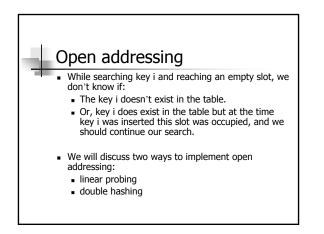


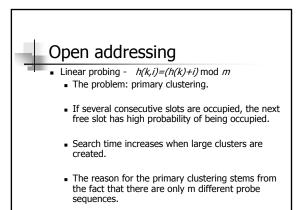


- There are two approaches to handle collisions:
- Chaining.
- Open Addressing.
- Chaining:
 - Each entry in the table is a linked list.
 - The linked list holds all the keys that are mapped to this entry.
- Search operation on a hash table which applies chaining takes $O(1 + \alpha)$ time.

Chaining This complexity is calculated under the assumption of uniform hashing. Notice that in the chaining method, the load factor may be greater than one.

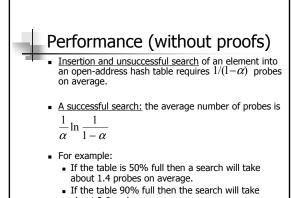






Open addressing

- Double hashing
 - $h(k,i)=(h_1(k)+ih_2(k)) \mod m$
 - Better than linear probing.
 - The problem h₂(k) can not have a common divisor with m (besides 1).
 - *m*² different probe sequences!



about 2.6 probes on average.

