.  
.  
.

**Double hashing example II – dealing with collision**

d(k) – how many spaces it goes if the current value is full.  
The hash keeps on checking in for eg k=31, at soace 5, then 9, then 0 (as d(k)=4, with a array length of 0-12).  
h(k) – the array index that the number starts at.  
  
Rehashing is quite an expensive exercise, so it is important to make an array of the correct size.

Complexity calculations are heavily effected by how this is implemented.

**Performance of Hashing**

O(n) worst case for: searches, insertions, removals on hash table.  
-The worst case when al keys inserted into map collide.  
Load factor: closer to 100, more issues happen – effects performance

The expected run time of all dictionary ADT operations in a hash table is O(1)

**In maps** keys are unique.  
However, In a dictionary, a key is not unique.