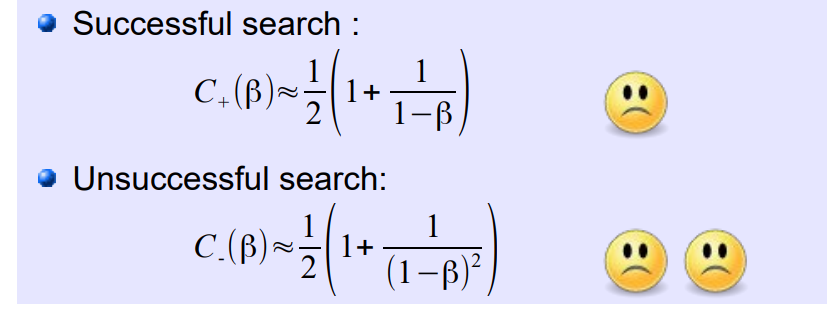
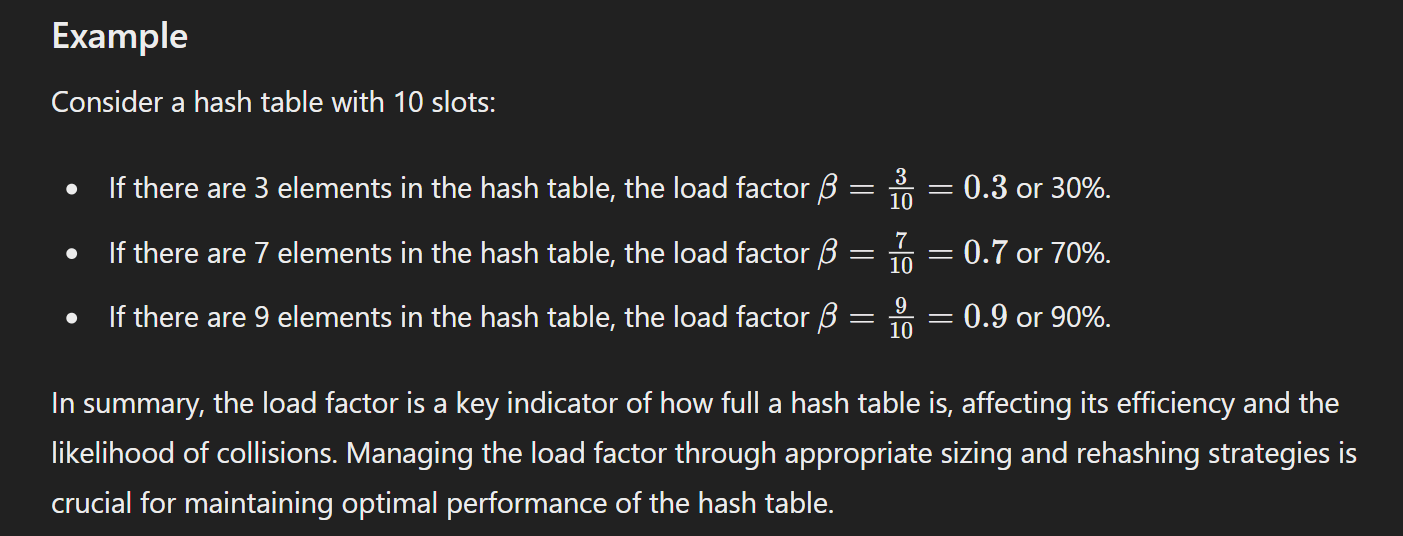
Number of probing steps with linear probing

The average probing required, depending on n the load factor ß<1 is given by the following formulas:  
  
A successful search means when the key being searched for is found the hash table, which in practice means that the search operations locates the desired key and can then return the associated value sorted in the hash table and unsuccessful search simply means that the key was not found in the hash table  
  
For example, if we are searching for the key “K” in the hash table and “K” is there, the search is considered to be successful

  
  
  
So, in the case when the table is nearly full, ß approaches to 100% which means that the number of probing steps required for a successful search increase significantly. As a result, the search time increases, making the HASH TABLE LESS EFFICIENT  
  
**The load factor (ß):** The load factor measure of how full a hash table is. It is defined as the ratio of the number of elements(keys) stored in the has table to the total number of slots(buckets) available in the table, which means:  
  
ß = Number of elements in the has table/Total number of slots in the hash table

1. **Low Load Factor**:
   * When the load factor is closer to 0
   * Many Empty slots, and collisions are less likely to occur
   * Search, insert and delete operations are generally more efficient because there are fewer collisions and less probing required
2. **High Load Factor**
   * When the factor is high, approaching to 1
   * Means that the hash table is nearly full, with most slots occupied
   * Search, insert and delete operations becomes less efficient due to the increased collisions and probing steps



Background knowledge for the next concept:  
  
A screenshot of a computer screen

Description automatically generated

**What is the reason for large increase in probing steps when the load factor becomes high?  
The reason for this is the effect of so-called primary clustering.**

The reason for this is the effect of so-called (primary) clustering. A primary cluster is understood to be a contiguously occupied sections in the table (for example, in the previously considered example, a cluster has formed from address 3 to 6). If the load factor increases, i.e. more and more entries in the table are filled, smaller clusters gradually melt together into longer and longer clusters.  
A diagram of a load factor

Description automatically generated

Linear Probing causes the effect of primary clusitering, which leads to long run times when the load factor becomes high. One way to avoid the problem of the primary clustering is to use other probing sequences: **Quadratic Probing**

A close-up of a text

Description automatically generated  
A math problem with numbers and equations

Description automatically generated

**The effect of primary clustering is avoided by quadratic probing. However, there is a second, less strong effect that has a negative impact on search times with increasing load factor, the so-called secondary clustering.  
  
A diagram of a clustering diagram

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer program

Description automatically generated**