Data Structures

**Hashing:**

* 1. Array
  2. Linked list
  3. BST (best)
  4. Direct Access Table – array index will be phone number directly.

Hashing is improvement. We can create hash function like hash(x)%S, and probe into the corresponding index of the hash table.

Int \*hash\_table[n]; Every element has pointer to a data structure containing the information.

Aim is to generate a unique key. The information needed will be found in the address the key element points to.

Chaining: one slot can have linked list associated with it.

Open addressing: keys are mapped in the same table.

If hash(x)%S is already filled with a key, then

1. Linear probing: insert at next free slot. [hash(x)+i]. clustering problem occurs.
2. Quadratic probing: insert at [hash(x)+i\*i] slot.
3. Double probing: insert at [hash(x)+i\*hash2(x)].

Chaining uses extra links for keys, whereas open addressing uses the space inside the same array.

Learn about cache performance of both and load factor on Wiki.

How is hashing implemented in programming.

What are examples of good hash functions.

**Pattern searching:**

**KMP –** if a part of pattern matches with a part of text, and the matched part of the pattern has a suffix which is also a prefix, then comparison begins with the next letter after the prefix and the next letter of the letter which was last matched in the text. This way the order becomes O(n), where n is the length of the text.

**Rabin-Karp –** it involves keeping hash of the pattern and comparing it with each window in the text.

When you use free you are actually telling the computer that you don't need that space anymore, so it marks that space as available for other data.

The pointer still points to that memory address. At this point that same space in the heap can be returned by another malloc call. When you invoke free a second time, you are not freeing the previous data, but the new data, and this may not be good for your program

**Trie:**

TrieNode

|  |  |
| --- | --- |
| Pointer to next struct | Bool isLeaf |

A pointer points to a struct containing 26 Trie nodes.

O(length of string) time to find the string in a Trie.