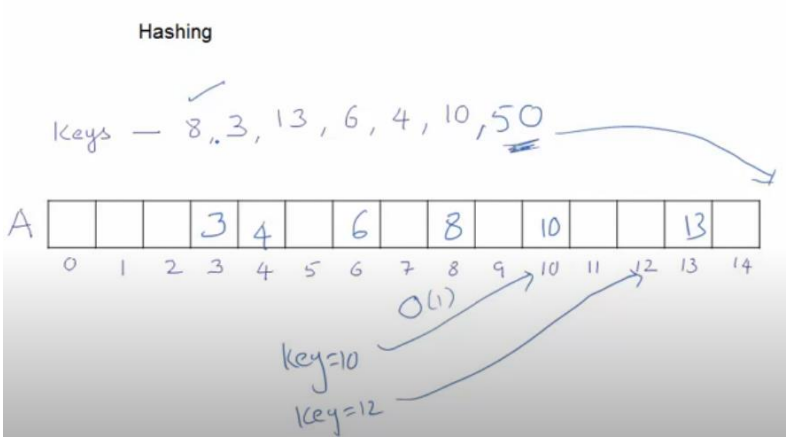
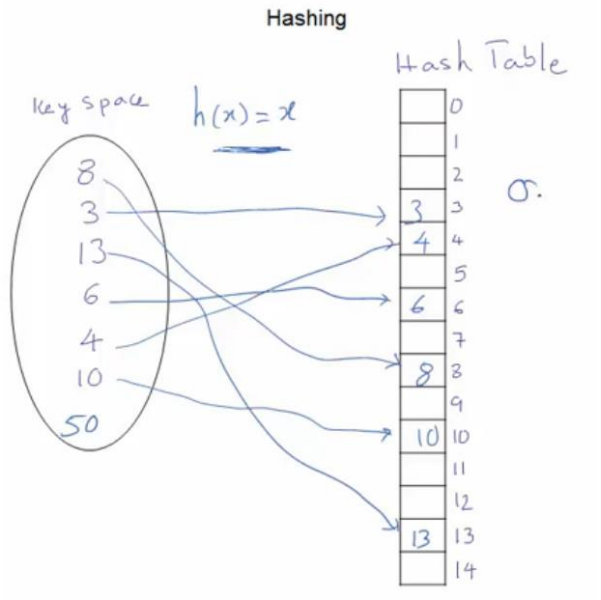


Burada 50 sayisini taa 50 arraya koymamiz gerekirse cok ciddi bir alan sikintisi yaratiriz Arrayin kapladigi alan artar.Bunun icin fonksiyonlara dayanan bir matematik formulu var



There are two type of function. One to one

H(x)=x is one to one.

And many to one.

H(x)=x burada mapping alices'dan sorumlu.we should modify the function to use efficiently using table.

Hash Function

h(x) = x % 10

Mod alimi yaparak secim yapacagiz bir sonraki resime gidelim

Bu ornekte size of hash table 10 olarak alinmis farkli da olabilir. Burada h(x) % 10 mod alimi yapacagiz. Ornek olarak 8 mod 10 bize 8 I veriyor dolayisiyla 8'e gidecek. 13 mod 10 ise collison oluyor. Birden fazla elementin ayni indexe map olmasi hash table'da mumkun. Bu collision olarak bilinioyr.bunun icin

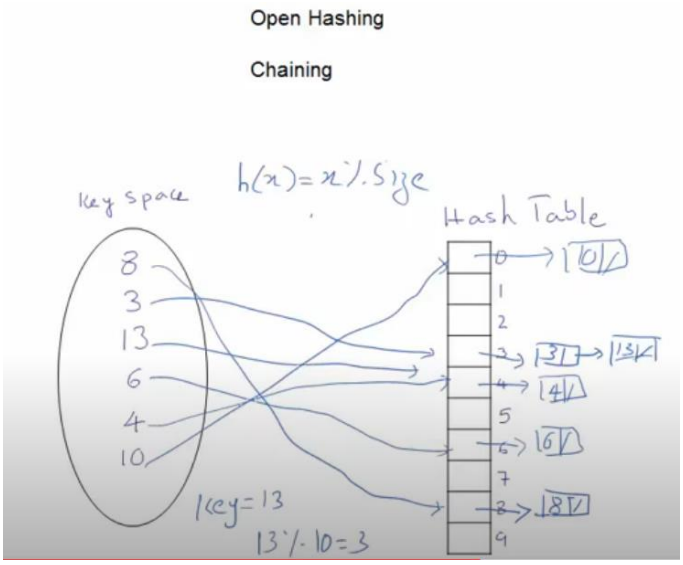
Collision Resolution Methods

1. Open Hashing
   Chaining

2. Closed Hashing

   Open Addressing
   1. Linear Probing
   2. Quadratic Probing

farkli cozumlerimiz var bakalim.

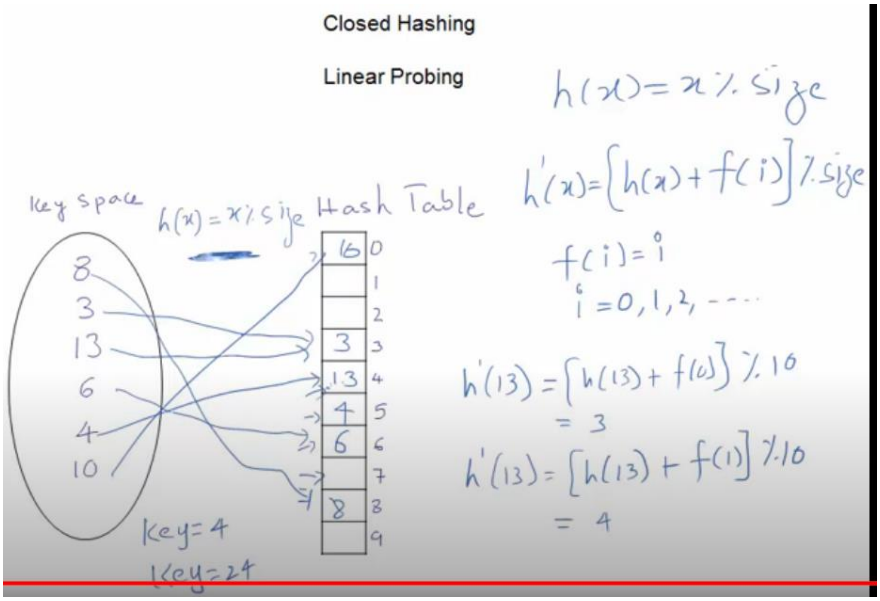


Chaining first method of removing collison

When we map any key while using h(x)=n % size function

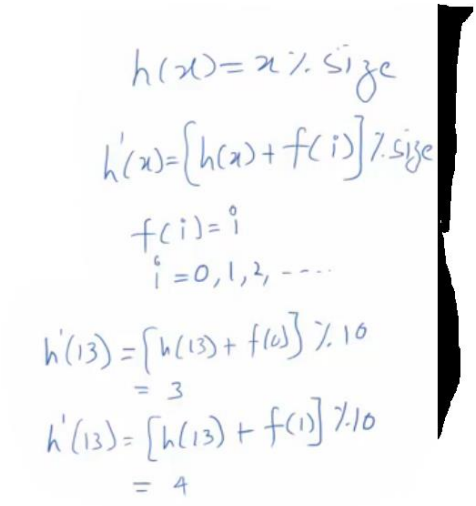
We add a new chain and we inser a new element 3 ve 13 burada ayni indexte cozum olarak we insert a key element at the same index in a chain

When I want to search a key element 13 mod 10 is 3



Linear probing; Bu methodta mesela 3 ve 13 cakisiyor dolaisiyla 13 3'un bir sonraki indexine gidiyor

Matematiksel aciklamasi; Burada 13 mode 10 3'uncu indexe geldi ama dolu sonra +1 alarak mode 10'a giriyor sonrasinda 4 indexi geldi eger 4 indexi doluysa

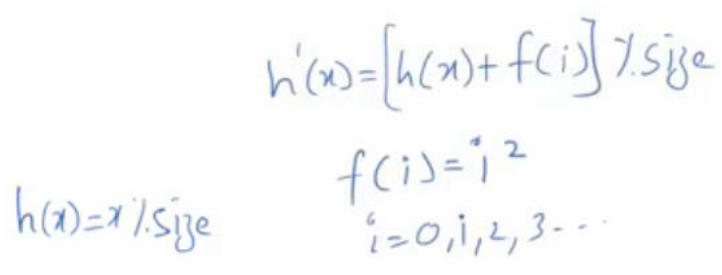


Bir sonraki islem f(2) oluyor yani her zaman bir sonraki indexe bakiyor ve arkasindaki olan matematik bu.

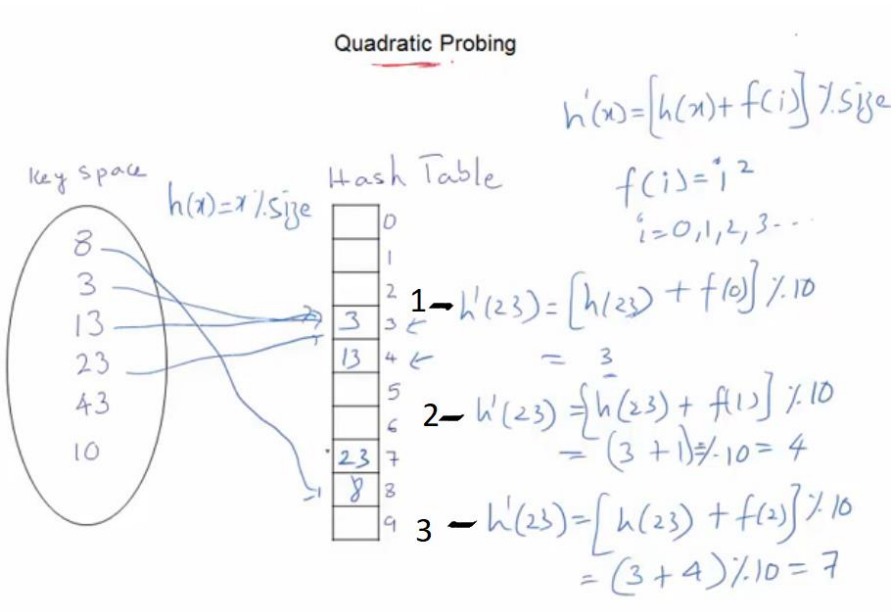
Temel mantik mod alma seklinde meselq 6 mod 10 kac yapiyor netten hesapla 6 dolayisiyla 6'inci indexe gidiyor. MeselA 13 mod 10 ile 3 mod 10 ayni indexte 13 mod 10 kendine bir index bulana kadar gitmeye devam ediyor .

Problem of linear probing is it causes clustering of elements bunch of elements will be at one single place at consecutive location they will be located continuously so this may cause a lot of time for searching an element so to avoid clustering we introduce

Quadratic Probing



Formul yine ayni ama soldaki gibi islem uyguluyoruz.Burada fark I'nin kare olmasi



Simdi 23'un mode 10 aldik alirken F 0 karesini alirsan yine 0 sonuc olarak 3 cikti.3'uncu index dolu yine islem yaptik busefer ikincide f 1 karesini alirsan yine 1 sonuc olarak Index 4 cikti ama orasi da dolu.Sonrasinda F 2 iken karesi 4 oldu 23 mode 10 3 sonuc olarak 3 uncu islemde parentew ici 3+4 mode 7 oldu sonucumuz 7 cikti.Sonrasinda 7'inci yere sayimizi yerlestirdik.

Burada elementler biraz birbirinden uzak ayri daha acik daha ferah collision olmuyor. Bu clustering of elementsi avoid ediyor yani engelliyor. And improves the searching time.