11.2-1

Suppose we use a hash function h to hash n distinct keys into an array T of length m. Assuming simple uniform hashing, what is the expected number of collisions? More precisely, what is the expected cardinality of ffk; lg W k ¤ l and h.k/ D h.l/g?

Sol)

no of collisions is the sum of no. of collisions for each possible smallest elemnt in the collision.

het di be the no
ki i
Problem 1
We assume simple unforss hashing
het our keys be esdered & kike kn?
of ls sit, eski and hide

11.2-5)

Suppose that we are storing a set of n keys into a hash table of size m. Show that if the keys are drawn from a universe U with jUj > nm, then U has a subset of size n consisting of keys that all hash to the same slot, so that the worst-case searching time for hashing with chaining is ‚.n/.

Sol)

In chaning we put all the elements in a same slot in a linked list.

nt.
a c
เว่า7.
centou
a Lunsd ust of all the ksy
Bach bash. toble ist T
1

ato xut eleme
hat has to j, i thove ase no utith
CHAINED- HASH SEARCH (T, h)
CHAINED-HASH-DELETE IT, m)
is meltnisenct intabl

11.3-1)

Suppose we wish to search a linked list of length n, where each element contains a key k along with a hash value h.k/. Each key is a long character string. How might we take advantage of the hash values when searching the list for an element with a given key?

Sol)

Each key is a long character thus to compare keys, at every node we need to perform a  
string comparison operation which is very time consuming. Instead we generate a hash  
value for the key (i.e., generate a numeric value for each string) we are searching for and  
comparing hash values h(k) along the length of the list, which turns out to be numeric  
values and the comparison is faster.

