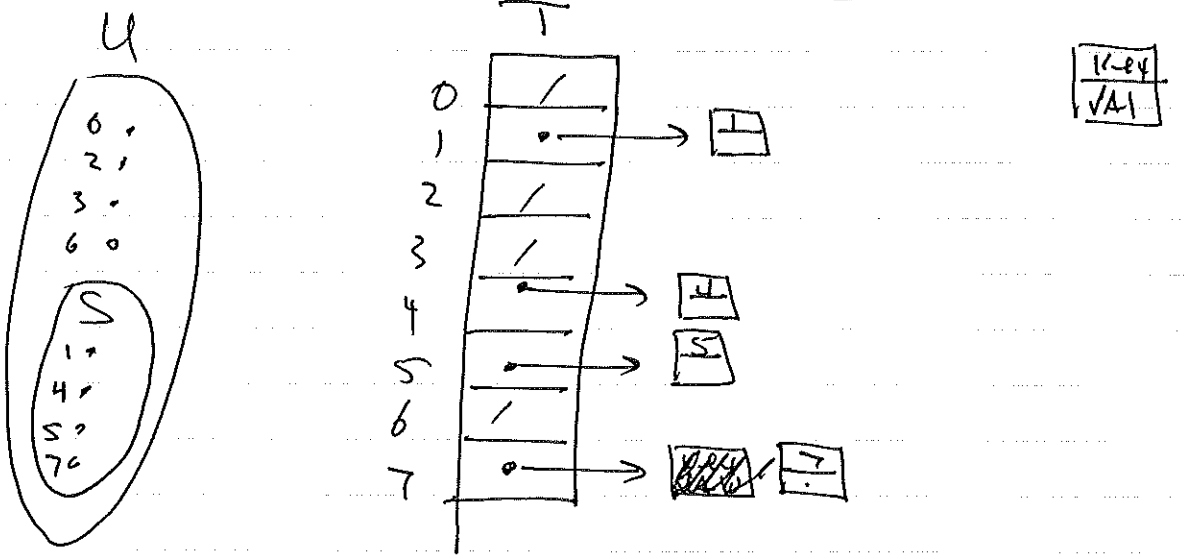
HASH TABLES (CHAP 13, SEC 2)

$U = \{$ ALL POSSIBLE KEYS $\}$

$S = \{$ KEYS ACTUALLY IN USE $\}$

$n = \text{length}[T]$.

EX $n=7$ DIRECT ADDRESS TABLE



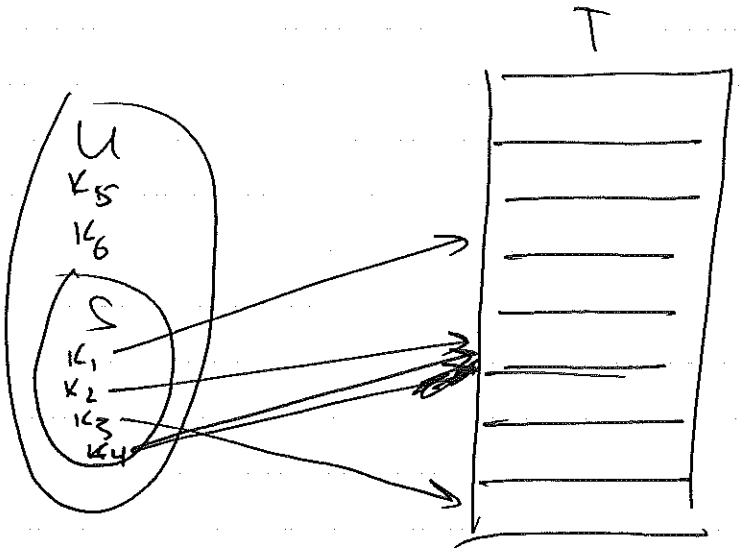
EX HASH TABLE: STORE (K,V) IN $T[h(K)]$ WHERE

$$h: U \to \{0, \ldots, n-1\}$$

IS A HASH FUNCTION.

NOTE $n < |U|$ SO h IS NOT ONE TO ONE. THUS CAN HAVE COLLISIONS

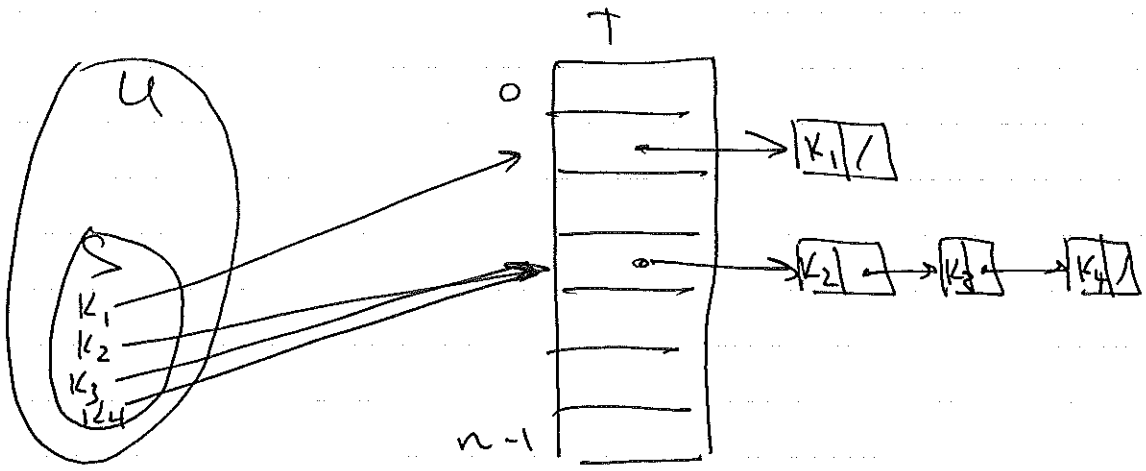
$$h(K_1) = h(K_2)$$



h SHOULD BE CHOSEN SO AS TO MINIMIZE THE PROBABILITY OF COLLISIONS, i.e. h 'DISTRIBUTES' THE PROB. OF COLLISION EVENLY ACROSS ALL INDICES $\{0, \ldots, n-1\}$.

∴ h IS DESIGNED TO LOOK 'RANDOM' ON BOTH SIDES, HENCE NAME 'hash'.

CHAINING: MULTIPLE KEYS INTO LINKED LIST.



Insert: Place new node at head of list $T[h(k)]$

Delete: Delete node from $T[h(k)]$.

Lookup: Do linear search of list $T[h(k)]$.

Defn: The <u>load factor</u> of a hash table is

$$\alpha = \frac{\#\text{Keys in use}}{\text{TableSize}} = \frac{|S|}{n}$$

(i.e. α is the average # of keys stored in any list, for chaining.

The efficiency of a collision res. method is analyzed in terms of α.

In Collision Res. By chaining we have for

- Unsuccessful search:

  $$\#\text{Comparisons} = \alpha$$

- Successful search

  $$\#\text{Comparisons} = 1 + \frac{\alpha}{2}$$

  ↑ other list not empty

  ↖ on avg. target is halfway down list.

SIMPLE UNIF. HASHING (SUH)

A RANDOM key is EQUALLY likely TO HASH INTO ANY SLOT: $\text{Prob}(h(k)=i) = \frac{1}{n}$ for $0 \le i \le n-1$.

IN THE LANGUAGE OF PROB. THEY U is THE SAMPLE SPACE AND

$$h: U \to \{0, \ldots, n-1\}$$

is A DISCRETE RANDOM VARIABLE WHICH is UNIFORMLY DISTRIBUTED, i.e. THE PROB. DENSITY FUNCTION FOR h is THE CONSTANT $1/n$.

NOTE WE DO NOT ASSUME THAT ANY TWO KEYS ARE EQUALLY LIKELY TO BE IN USE, SOME KEYS MAY BE MORE LIKELY THAN OTHERS. THUS TO CONFORM TO S.U.H., h MUST BE CHOSEN WITH THE DISTRIBUTION OF KEYS IN MIND.

EX. IF KEYS ARE UNIF. DIST. OVER $U = \mathbb{Z}$, THEN $h(k) = k \bmod n$ SATISFIES S.U.H.

EX IF KEYS ARE UNIF. DIST. OVER $0 \le k < 1$ i.e. $U = [0, 1)$, THEN $h(k) = \lfloor km \rfloor$ SATISFIES SUH.

OBSERVE $h: \left[\frac{i}{n}, \frac{i+1}{n}\right) \to \{i\}$ $\quad (0 \le i \le n-1)$

IF THE PROB. DIST. OF KEYS IS NOT KNOWN WE MUST TAKE A HEURISTIC APPROACH TO CHOOSING h.

ASSUME FROM NOW ON: $U = \mathbb{N} = \{0, 1, 2, \ldots\}$

GO TO P. 96.

Ex Insert Keys

17 33 29 13 12, 44

45 57 9 3 81, 90

19 97 35

into hash table of length = 11

- Resolve coll by chaining $h(k) = k \bmod 11$
- determine α, # collisions
- Resolve coll by open addressing.

$$h(k,i) = (k \bmod 11 + 3i) \bmod 11$$

D. 100 in C

ASSUMAR FOR SAKE OF SIMPLICITY THAT TABLE ENTRIES ARE KEYS WITH NO SATELLITE DATA (i.e. NO 'value').

TO INSERT key K INTO T DO

```
fail = 1;
i = 0;
while (i < n && fail) {
    j = h(K, i);
    if (T[j] == EMPTY) {
        T[j] = K;
        fail = 0;
    } else {
        i++;
    }
}
if (fail) return error;
else return j;
```

say #define ... -1

TO Lookup for key K in T DO:

i = 0;

~~[illegible]~~

101 in C

```
found = 0;
i = 0;
do {
    j = h(k, i);
    if T[j] = k
        found = 1;
    else
        i++;
} while ( i < n && !found );

if (!found) return EMPTY;
else return j;
```