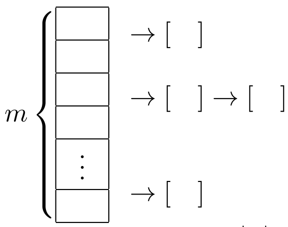
Lecture 10

Hashing



m slots, universe u, $|u| > m$, n entries
$h(k)$ hash function, $h(k) = h(k') \rightarrow$ collision $\rightarrow$ chaining!
chaining $\rightarrow$ linked list in each bucket

$$E[T] = \alpha = \frac{n}{m}$$

$$E[T] \rightarrow \text{ consider searching for } x \text{ chosen at random uniformly from items in } T$$

Number of elements examined in search for x
$= 1 + \#$ of elements in fron of x in chain
$= 1 + \#$ of elements in x's bucket inserted later than x

Let $x_1, x_2, x_3, \ldots, x_n$ be the elements in insertion order. Let

$$x_{i,j} = \begin{cases} 1 & h(x_i) = h(x_j) \\ 0 & \text{otherwise} \end{cases}$$

$$\begin{aligned} E[T] &= E\left[\frac{1}{n}\sum_{i=1}^{n}\left(1 + \sum_{j=i+1}^{n} X_{i,j}\right)\right] \\ &= \frac{1}{n}\sum_{i=1}^{n}\left(1 + \underbrace{\sum_{j=i+1}^{n} \underbrace{E[X_{i,j}]}_{1/m}}_{(n-i)/m}\right) \\ &= 1 + \sum_{i=1}^{n} \frac{n-i}{m} \\ &= 1 + \frac{\alpha}{2} - \frac{\alpha}{2n} \end{aligned}$$

Properties we would like:

- not all go to same bucket
- spread out keys, no empty buckets
- need deterministic
- efficient
- use all bits of key

Non-numeric keys

"help" $\rightarrow$ big number
"pleh"

Division Method: $k \bmod m$
Multiplication Method:

$$h(k) = \lfloor m \cdot \text{ frac}(k \cdot A) \rfloor \text{ where } A \in (0, 1]$$

Open Addressing

Instead of using chaining, we can store all the items directly in T. When a given bucket is full, we have a rule for determining which bucket to use next. The rule must be deterministic so that we can check the same sequence of buckets when we are searching for the item later.

$h(k)$
$h(k'), h(k') + 1, h(k') + 2, \ldots \rightarrow$ probe sequence
linear probe sequencing $\rightarrow$ clustering
Use non-linear probing!
Use double hashing!