Separate Chaining

Have a list referenced by each slot of hash table that holds all elements that hash to that slot (one hash function).

\[ e \rightarrow \text{hash}(e) \]

**Insert** \( e \) add it to list table[\( \text{hash}(x) \)]

**Locate** \( e \) search within list table[\( \text{hash}(x) \)]

**Remove** \( e \) remove \( e \) from list table[\( \text{hash}(x) \)]
Resizing hash table

No absolute limit on \( n/m \) (could go arbitrarily high) but cost is too high

Open addressing \( \leq 1 \)

Resize upward when \( L \) reaches \( 2^k \)
Resize downward when \( L \) reaches \( 2^{k-1} \)
Analysis

Expected cost for unsuccessful search

every reference we follow

is a “probe”

\[
\text{hash}(k) = 1, 2, 3, \text{full}, 4
\]

expected list length

\[
\frac{n}{m} = \alpha
\]

\[
E\left[\text{# probes in an unsuccessful search}\right] = 1 + \alpha
\]

\[
E\left[\text{# probes in a successful search}\right] = 1 + \frac{\alpha}{2} - \frac{2}{2n}
\]

open addressing

\[
\frac{1}{1-\alpha} = 1 + \alpha + \alpha^2 + \alpha^3 + \ldots
\]