Tradeoff between space + search time

as we increase α_*

use less space but higher expected cost per search

this is a pretty large value

$\alpha_* = 7/8$ $\quad \frac{1}{1-7/8} = 8$

As we decrease α_*, use more space but have lower expected cost per search

$\alpha_* = 1/4$ $\quad \frac{1}{1-1/4} = 4/3$ $\quad$ space $4m$

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.hashCode()

insert(e) add it to list table[hash(x)]

locate(e) search within list table[hash(x)]

remove(e) remove e from list table[hash(x)]

Resizing hash table

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

Open addressing $\alpha < 1$

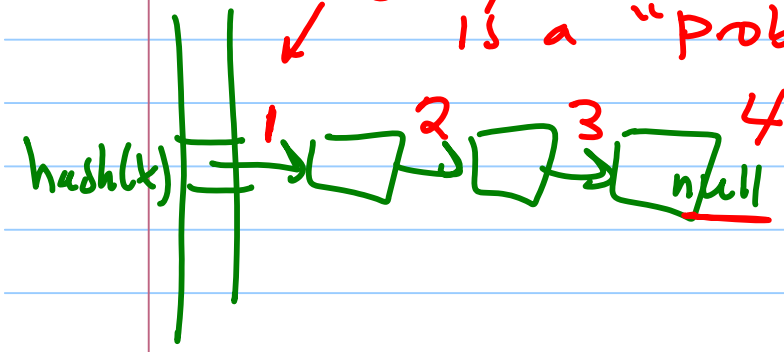
resize upward when α reaches $2\alpha_*$

resize downward when α reaches $\alpha_*/2$

Analysis

Expected cost for unsuccessful search

every reference we follow is a "probe"



expected list length

$$\frac{n}{m} = \alpha$$

$$E\left[\begin{array}{c}\#\text{ probes in an} \\ \text{unsuccessful search}\end{array}\right] = 1 + \alpha$$

open addressing

$$\frac{1}{1-\alpha} = 1 + \alpha + \alpha^2 + \ldots$$

$$E\left[\begin{array}{c}\#\text{ probes in an} \\ \text{successful search}\end{array}\right] = 1 + \frac{\alpha}{2} - \frac{\alpha}{2n}$$

Summary of Set ADT Implementations

$m = |U|$

element ref + next ref

slots marked as deleted

hash table

elements in set

only good $n > m/4$

Data Structure	Unsuccessful Search	Successful Search	Approximate Space Usage
Direct Addressing	1	1	m
Separate Chaining	$1+\alpha$	$1+\frac{\alpha}{2}-\frac{\alpha}{2n}$	$2n+m=n\left(3+\frac{1}{\alpha}\right)$
Separate Chaining (α=1/2)	1.5	≈ 1.25	$4n$
Separate Chaining (α=3/4)	1.75	≈ 1.375	$3\frac{1}{3}n$
Separate Chaining (α=1)	2	≈ 1.5	$3n$
Separate Chaining (α=3)	4	≈ 2.5	$2\frac{1}{3}n$
Open Addressing	$\frac{1}{1-\alpha}$	$\frac{1}{\alpha}\ln\frac{1}{1-\alpha}$	$m=\frac{(n+d)}{\alpha}$
Open Addressing (α=1/4)	4/3	≈ 1.15	$4(n+d)$
Open Addressing (α=1/2)	2	≈ 1.39	$2(n+d)$
Open Addressing (α=3/4)	4	≈ 1.85	$\frac{4}{3}(n+d)$

$d=0$

$\alpha = n/m$

$m=n$

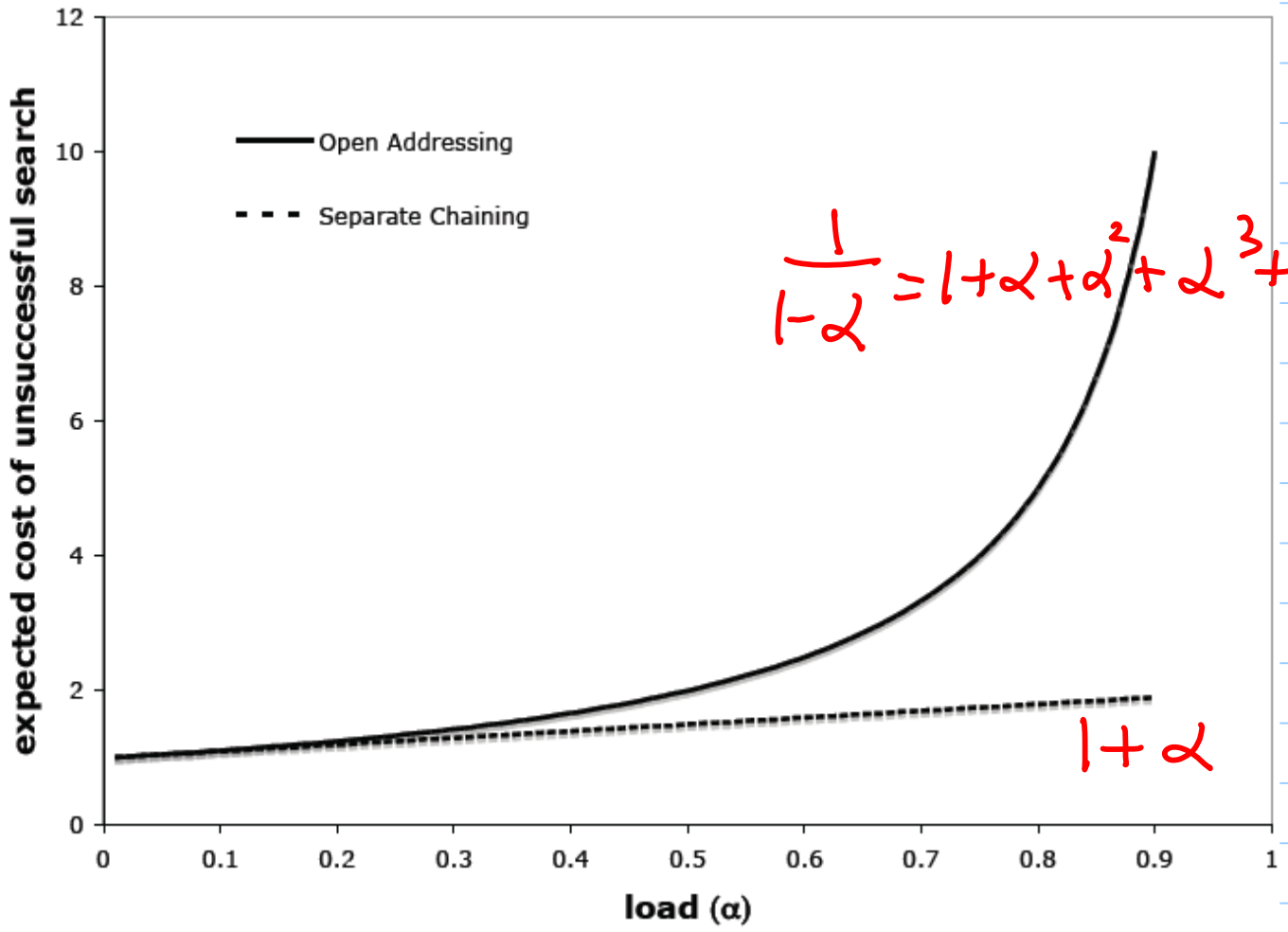
$m=2(n+d)$

$\alpha = \frac{n+d}{m}$

hash table size

Unsuccessful Search Cost as a Function of α

α = load



$\frac{1}{1-\alpha} = 1+\alpha+\alpha^2+\alpha^3+\cdots$

$1+\alpha$

Comparison of Search Cost for Space Usage of ~3n