Tradeoff between space and search time

As we increase $\lambda^*$, use less space but higher expected cost per search.

This is a pretty large value $\lambda^* = 7/8 \quad \frac{1}{1 - 7/8} = 8$

As we decrease $\lambda^*$, use more space but have lower expected cost per search.

$\lambda^* = 1/4 \quad \frac{1}{1 - 1/4} = 4/3 \quad \text{space} \quad 4m$
Separate Chaining

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

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

\[
\text{locate}(e) \quad \text{search within list table}[\text{hash}(x)]
\]

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

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

Open addressing \( \lambda < 1 \)

Resize upward when \( \lambda \) reaches \( 2^k \)
Resize downward when \( \lambda \) reaches \( \lambda^* / 2 \)
**Analysis**

Expected cost for unsuccessful search

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

expected list length

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

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

open addressing

\[ \frac{1}{1 - \alpha} = 1 + \alpha + \alpha^2 + \ldots \]
# Summary of Set ADT Implementations

<table>
<thead>
<tr>
<th>Data Structure</th>
<th>Unsuccessful Search</th>
<th>Successful Search</th>
<th>Approximate Space Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Direct Addressing</td>
<td>1</td>
<td>1</td>
<td>( m )</td>
</tr>
<tr>
<td>Separate Chaining</td>
<td>( 1 + \alpha )</td>
<td>( 1 + \frac{\alpha}{2} - \frac{\alpha}{2n} )</td>
<td>( 2n + m = n \left( 3 + \frac{1}{\alpha} \right) )</td>
</tr>
<tr>
<td>( \alpha = 1/2 )</td>
<td>1.5</td>
<td>( \approx 1.25 )</td>
<td>( 4n )</td>
</tr>
<tr>
<td>( \alpha = 3/4 )</td>
<td>1.75</td>
<td>( \approx 1.375 )</td>
<td>( 3 \frac{1}{3} n )</td>
</tr>
<tr>
<td>Separate Chaining</td>
<td>2</td>
<td>( \approx 1.5 )</td>
<td>( 3n )</td>
</tr>
<tr>
<td>( \alpha = 1 )</td>
<td>4</td>
<td>( \approx 2.5 )</td>
<td>( 2 \frac{1}{3} n )</td>
</tr>
<tr>
<td>Separate Chaining</td>
<td>2</td>
<td>( \approx 1.5 )</td>
<td>( 3n )</td>
</tr>
<tr>
<td>( \alpha = 3 )</td>
<td>4</td>
<td>( \approx 2.5 )</td>
<td>( 2 \frac{1}{3} n )</td>
</tr>
<tr>
<td>Open Addressing</td>
<td>( \frac{1}{1 - \alpha} )</td>
<td>( \frac{1}{\alpha \ln \frac{1}{1 - \alpha}} )</td>
<td>( m = \frac{(n + d)}{\alpha} )</td>
</tr>
<tr>
<td>( \alpha = 1/4 )</td>
<td>4/3</td>
<td>( \approx 1.15 )</td>
<td>( 4(n + d) )</td>
</tr>
<tr>
<td>( \alpha = 1/2 )</td>
<td>2</td>
<td>( \approx 1.39 )</td>
<td>( 2(n + d) )</td>
</tr>
<tr>
<td>( \alpha = 3/4 )</td>
<td>4</td>
<td>( \approx 1.85 )</td>
<td>( \frac{4}{3} (n + d) )</td>
</tr>
</tbody>
</table>
Unsuccessful Search Cost as a Function of Load

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

**Graph:**
- **Open Addressing**
- **Separate Chaining**

**Axes:**
- **Y-axis:** Expected cost of unsuccessful search
- **X-axis:** Load (\( \alpha \))

**Legend:**
- Open Addressing
- Separate Chaining

**Notes:**
- The graph compares the expected cost of unsuccessful search between Open Addressing and Separate Chaining.
- The formula \( \frac{1}{1-\lambda} = 1 + \lambda + \lambda^2 + \lambda^3 + \ldots \) illustrates the relationship between load and expected cost.
Comparison of search cost for space usage of ~3n.