Running Time Observations:

<table>
<thead>
<tr>
<th>Collision Resolution Strategy</th>
<th>Successful</th>
<th>Unsuccessful</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear Probing</td>
<td>(\frac{1}{2}(1 + \frac{1}{1-\alpha}))</td>
<td>(\frac{1}{2}(1 + \frac{1}{1-\alpha})^2)</td>
</tr>
<tr>
<td>Double Hashing</td>
<td>(\frac{1}{\alpha} \ln(\frac{1}{1-\alpha}))</td>
<td>(\frac{1}{1-\alpha})</td>
</tr>
</tbody>
</table>

ReHashing:
What happens when the array fills?

...or a better question:

Algorithm:

Which collision resolution strategy is better?
- Big Records:
- Structure Speed:

What structure do hash tables replace?

What constraint exists on hashing that doesn’t exist with BSTs?

Why talk about BSTs at all?
Implementation of _______________

<table>
<thead>
<tr>
<th>insert</th>
<th>removeMin</th>
<th>Implementation</th>
</tr>
</thead>
<tbody>
<tr>
<td>O(n)</td>
<td>O(n)</td>
<td>Unsorted Array</td>
</tr>
<tr>
<td>O(1)</td>
<td>O(n)</td>
<td>Unsorted List</td>
</tr>
<tr>
<td>O(lg(n))</td>
<td>O(1)</td>
<td>Sorted Array</td>
</tr>
<tr>
<td>O(lg(n))</td>
<td>O(1)</td>
<td>Sorted List</td>
</tr>
</tbody>
</table>

Q1: What errors exist in this table? (Fix them!)

Q2: Which algorithm would we use?

Implementing a (min)Heap as an Array

Operations:
leftChild(index) :=
rightChild(index) :=
parent(index) :=

Insert:

A New Tree-like Structure:

A complete binary tree T is a min-heap if:
  •
  •

CS 225 – Things To Be Doing:
1. Theory Exam 3 starts next week (Thursday, November 8th)
2. MP5 EC+7 deadline is today – earn the extra credit!
3. lab_hash released Wednesday
4. Daily POTDs are ongoing!