Fundamental Algorithms 7 - Solution Examples

Exercise 1 (Hash Function)
Let \( n = 1000 \). Compute the values of the hash function \( h(k) = \lfloor n(ak - \lfloor ak \rfloor) \rfloor \) for the keys \( k \in \{61, 62, 63, 64, 65\} \), using \( a = \frac{\sqrt{5} - 1}{2} \). What do you observe?

Solution:

<table>
<thead>
<tr>
<th>( k )</th>
<th>61</th>
<th>62</th>
<th>63</th>
<th>64</th>
<th>65</th>
</tr>
</thead>
<tbody>
<tr>
<td>( h(k) )</td>
<td>700</td>
<td>318</td>
<td>936</td>
<td>554</td>
<td>172</td>
</tr>
</tbody>
</table>

The hash function is “non-smooth”: similar entries lead to different hash values.

Exercise 2 (Hash Table)
Let \( T \) by a hash-table of size 9 with the hash function \( h : U \to \{0, 1, \ldots, 8\}, k \mapsto k \mod 9 \). Write down the entries of \( T \) after the keys 5, 28, 19, 15, 20, 33, 12, 17, and 10 have been inserted. Use chaining to resolve collisions.

Solution:

\[
\begin{array}{cccccccccc}
i & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\
\end{array}
\]

The \( [] \)-notation denotes the lists that are stored in each hash table slot.

Exercise 3 (Open Hash)
Now, let \( T \) be a hash table of size 11, using open addressing with the following hash functions

1. \( h(k, i) := (k + i) \mod 11 \)
2. \( h(k, i) := (k \mod 11 + 2i + i^2) \mod 11 \)
3. \( h(k, i) := (k \mod 11 + i \cdot (k \mod 7 + 1)) \mod 11 \)

Insert the keys 5, 19, 27, 15, 30, 34, 26, 12, and 21 (in that order) and state which keys require the longest probe sequence in the resulting tables.

Solution:

1. Linear probing:

\[
\begin{array}{cccccccccc}
i & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\
T[i] & 34 & 12 & 15 & 5 & 27 & 26 & 19 & 30 & 21 \\
\end{array}
\]

Longest probe sequence is 4 (for 26).

2. Quadratic probing:

\[
\begin{array}{cccccccccc}
i & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\
T[i] & 30 & 34 & 27 & 15 & 5 & 26 & 19 & 12 & 21 \\
\end{array}
\]
Longest probe sequence is 2 (for 27 and 12).

3. Double hashing probing:

<table>
<thead>
<tr>
<th>i</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T[i]$</td>
<td>30</td>
<td>27</td>
<td>12</td>
<td>21</td>
<td>15</td>
<td>5</td>
<td>34</td>
<td>19</td>
<td>26</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Largest probe sequences is 5 (for 34 and 21).

*Note:* Contrary to this example, double hashing usually beats linear or quadratic probing. Moreover, using a larger table for open addressing is recommended.

**Exercise 4 (Hashing the Universe)**

Consider a universe $U$ of keys, where $|U| > mn$, and a hash function $h : U \rightarrow \{0, 1, \ldots, n - 1\}$. Show that there are at least $m$ elements of $U$ which are mapped to the same hash value, i.e. there is a subset $A$ of $U$ with $|A| = m$ and $h(a_1) = h(a_2)$ for all $a_1, a_2 \in A$.

**Solution:**

Assume the opposite, i.e. that for all $n$ values of the hash function the number of elements in $U$ that are hashed to this value is smaller than $m$. As a consequence, the number of elements that are hashed to any of the $n$ keys is smaller than $nm$. This contradicts the fact that $U$ is considered to have more than $nm$ elements. Hence, our assumption has to be false, and there has to be at least one subset containing at least $m$ elements.