Computer Engineering II
Exercise Sheet 6

Quiz

1 Quiz

a) What happens if a hash function is biased to favor some buckets?

b) What do we need to take into account to analyze the time complexity of using a hash table that picks hash functions from a universal family?

c) Is hashing a good idea if you need every single insert/delete/search to be fast?

Basic

2 Trying out hashing

Let $N = \{10, 22, 31, 4, 15, 28, 17, 88, 59\}$ and $m = 11$. Let $h_1(k) = k \mod m$; now build three hash tables: one for linear probing with $c = 1$, one for quadratic probing with $c = 1$ and $d = 3$, and one for double hashing with $h'(k) = 1 + (k \mod (m - 1))$. Reminder:

- Linear probing: $h_i(k) \equiv h(k) + ci \mod m$
- Quadratic probing: $h_i(k) \equiv h(k) + ci + di^2 \mod m$
- Double hashing: $h_i(k) \equiv h(k) + ih'(k) \mod m$

Note: You can just do half the exercise in class and the rest at home since it is somewhat time consuming. Also, don’t give up if a probing sequence seems to go on for too long!

3 Using hash tables

Assume you are given two sets of integers, $S = \{s_1, \ldots, s_q\}$ and $T = \{t_1, \ldots, t_r\}$.

a) Give an algorithm to check whether $S \subseteq T$ that uses hash tables.

b) What is the time complexity of your algorithm? Remember Quiz question c)!
4 r-independent hashing

Given a family of hash functions \( \mathcal{H} \subseteq \{ U \to M \} \), we say that \( \mathcal{H} \) is \textit{r-independent} if for every \( r \) distinct keys \( (x_1, \ldots, x_r) \) and every \( h \) sampled uniformly from \( \mathcal{H} \), the vector \( (h(x_1), \ldots, h(x_r)) \) is equally likely to be any element of \( M^r \).

a) Show that if \( \mathcal{H} \) is 2-independent, then it is universal. Hint: use that \( \mathcal{H} \) is universal if and only if \( \Pr[h(k) = h(l)] = \frac{1}{m} \) for keys \( k \neq l \).

b) Show that the universal family \( \mathcal{H} \) defined in the script (Theorem 6.9) is not 2-independent.

5 Obfuscated quadratic probing

Consider Algorithm 1 with \( m = 2^p \) for some integer \( p \).

Algorithm 1 Obfuscated quadratic probing: search

\[ \begin{align*}
\text{Input:} & \quad \text{key } k \text{ to search for} \\
1: & \quad i := h(k) \\
2: & \quad \text{if } M[i] = k \text{ then} \\
3: & \quad \quad \text{return } M[i] \\
4: & \quad \text{end if} \\
5: & \quad j := 0 \\
6: & \quad \text{for } l \in \{0, \ldots, m-1\} \text{ do} \\
7: & \quad \quad j := j + 1 \\
8: & \quad \quad i := (i + j) \mod m \\
9: & \quad \quad \text{if } M[i] = k \text{ then} \\
10: & \quad \quad \quad \text{return } M[i] \\
11: & \quad \quad \text{end if} \\
12: & \quad \text{end for} \\
13: & \quad \text{return } \bot
\end{align*} \]

a) Show that this is an instance of quadratic probing by giving the constants \( c \) and \( d \) for a hash function \( h_i(k) = h(k) + ci + di^2 \).

b) Prove that the probing sequence of every key covers the whole table. Do this in two steps:

- Show that \( h_s(k) \equiv h_r(k) \mod m \) for \( r < s \) if and only if \((s-r)(s+r+1) = t2^{p+1}\) for some integer \( t \).
- Show that only one of \((s-r)\) and \((s+r+1)\) can be even, then show that \((s-r)(s+r+1) = t2^{p+1}\) has no solutions if \( r < s \) and \( r, s < m \).

6 Not quite universal hashing

Remember the universal family from the script: \( \mathcal{H} := \{ h_a : a \in [m]^{r+1} \} \) where \( h_a(u_0, \ldots, u_r) = \sum_{i=0}^{r} a_i \cdot u_i \mod m \). Show that if we restrict the \( a_i \) to be nonzero, then \( \mathcal{H} \) is no longer a universal family if \( r \geq 1 \).

Hint: How likely is it for \( x = (0, \ldots, 0) \) and \( y = (1, \ldots, 1) \) to collide?