Open-addressed hashing

Insert(x):

\[
\text{for } i = 0 \text{ to } m-1 \\
\text{if } T[h_i(x)] \text{ is empty} \\
\quad T[h_i(x)] \leftarrow x \\
\text{return }
\]

return FULL

\[h_0(x), h_1(x), ..., h_{m-1}(x) \text{ is a permutation of } 0, 1, ..., m-1\]

Linear probing:

\[h_i(x) = (h_0(x) + i) \mod m\]

Binary probing:

\[h_i(x) = \left( h_0(x) \oplus i \right) \mod m \quad (m=2^n)\]

\[h_0(x) = 01010\]
\[h_1(x) = 01011\]
\[h_2(x) = 01000\]
\[h_3(x) = 01001\]
\[h_4(x) = 01110\]
\]

Chained:

universal hashing

\[\Pr[h(x) = h(y)] \leq \frac{1}{m}\]

\[\Rightarrow O(1) \text{ lookup} \]
\[\Rightarrow O(1) \text{ exp. insertion/}
\text{ amortized deletion}\]

FICTION: \(\langle h_0(x), h_1(x), ... \rangle\) is a random permutation

every perm has prob. \(\frac{1}{m!}\)

\[T(n,m) = \# \text{ probes need to insert } n \text{ th item into a table of size } m\]

\[E[T(n,m)] \leq 1 + \frac{n}{m} E[T(n-1, m-1)]\]

Induction:

\[\frac{m}{m-n} = 2 = \frac{1}{1-\alpha}\]
Loose binary probing:

\[
\text{for } k = 0 \text{ to } \lg m \\
\text{if } B_k(x) \text{ is not full} \\
\text{put } x \text{ into } B_k(x) \\
\text{return }
\]

Time = \(O(\text{size of largest full block containing } h_0(x))\)

\(B_k(x) \) is full if all \(2^k\) slots are occupied

\(B_k(x) \) is popular if \(\# \exists y \mid \text{size } ho(y) \leq B_k(x) \geq 2^k\)

\(\text{popular } \Rightarrow \text{full}\)

\(B_k \text{ full } \Rightarrow B_k \text{ popular or sibling/uncle of } B_k \text{ is popular}\)

\[F_k = \Pr[B_k(x) \text{ is full}]\]

\[P_k = \Pr[B_k(x) \text{ is popular}]\]

\[F_k \leq P_k + \sum_{j \geq k} P_j\]

We need to understand \(P_k\) \((\text{fix } k)\)

\[Y = \# \exists y \mid \text{size } h_0(y) \leq B_k(x)\]

So \(P_k = \Pr[Y \geq 2^k]\)

\[E[Y] = \frac{1}{2} \cdot 2^k\]

assuming \(h_0\) is uniform

\(P_k = \Pr[Y \geq 2E[Y]]\)

assuming \(h_0\) is pairwise-independent

\[\Pr[Y \geq (1+\delta)E[Y]] < \frac{1}{8 \cdot E[Y]}\]

Cheb's \(\neq\)

Set \(\delta = 1\)

\[\Pr[Y \geq 2E[Y]] < \frac{1}{E[Y]} = \left[\frac{1}{2^{k-1}}\right]\]

\[\Rightarrow F_k = O(2^{-k})\]

\[E[\text{Time}] = \sum_{k=0}^{\lg m} O(2^k) \cdot F_k = \frac{\sum_{k=0}^{\lg m} O(2^k)}{O(1)} = O(\log m)\]
If hash values are \( 4 \)-independent

Chebyshev's Inequality
\[
\Pr\left[ |Y - E[Y]| \geq (1+\delta)\mu \right] = O\left(\left(\frac{1}{\delta^2\mu}\right)^2\right)
\]
\[
\Pr\left[ |Y - 2\cdot E[Y]| \geq 0 \right] = O(4^{-k}) \Rightarrow F_k = O(4^{-k})
\]
\[
E[\text{Time}] = \sum_{k=0}^{\infty} O(2^k) \cdot F_k = \sum_{k=0}^{\infty} O(2^{-k}) = O(1)
\]

We actually need **5-uniform** hashing

For all distinct \( v, w, x, y, z \)

For all \( h, i, j, k, l \)
\[
\Pr[h_0(v)=h \text{ and } \ldots \text{ and } h_0(z)=l] = \frac{1}{m^3}
\]

Carter Wegman
\[
h_0(x) = ((a + bx + cx^2 + dx^3 + ex^4) \mod p) \mod m
\]
where \( a, b, c, d, e \in \mathbb{Z}_p \)

Thorp Zang 2010:

Twisted tabulation:
\[
h(x, y) = A[x] \oplus B[y] \oplus C[x+y]
\]
5-uniform

Poțrașcu Thompson 2011:

Tabulation:
\[
h(x_1, \ldots, x_c) = A_1[x_1] \oplus A_2[x_2] \oplus \ldots \oplus A_c[x_c]
\]
3-uniform

still get \( O(1) \) whp
\( \ell \) depends on \( c \)