Lecture 8: Hash Tables with Linear Probing

Hash Tables with Linear Probing

We saw hashing with chaining. Using universal hashing we get expected $O(1)$ time per operation. One disadvantage is that chaining requires a list data structure at each bucket. Today we will discuss another popular technique called linear probing. We will mostly be following Kent Quanrud's thesis, which has nice figures and more detailed explanations, including historical notes. For this reason, we will be high-level in our description.

Linear Probing

Let the universe be U, with $|U| = u$. The size of the hash table is $A[0 \dots m-1]$. We pick a random hash function h from a hash family H.

Algorithm 1 insert(x)

$i \leftarrow h(x)$
while $A[i]$ is not empty **do**
    $i \leftarrow (i+1) \pmod m$
$A[i] \leftarrow x$

Algorithm 2 find(x)

$i \leftarrow h(x)$
while $A[i]$ is not empty **do**
    if $A[i] = x$ **then**
        return Yes
    $i \leftarrow (i+1) \pmod m$
return No

The `delete(x)` operation is more complicated. There are different strategies, but we want to maintain the correctness of `insert` and `find`. To delete x, we first find it. Say x is in $A[i]$. If we just set $A[i] =$ empty, we create a "hole". This could cause a subsequent `find` for an element whose probe sequence passed through i to fail incorrectly.

We must fill the hole. One strategy is to scan from i to the right. We look for an element y in the same block of occupied cells that can be moved to fill the hole at i. This process is repeated until the created hole can be filled by an empty cell.

Algorithm 3 delete(x)

$i \leftarrow$ location of x found by find(x)
if i is invalid **then return**
$A[i] \leftarrow$ empty
$j \leftarrow i$
loop
  $j \leftarrow (j+1) \pmod m$
  if $A[j]$ is empty **then**
    break
  Let $y = A[j]$
  Re-insert y starting from its hash position $h(y)$, effectively moving it if necessary to fill the original hole. This can be complex; a simpler (but correct) method is to re-insert all items in the run following the hole.

Note: The handwritten notes contain a more complex, potentially incomplete pseudocode for a specific hole-filling strategy. The algorithm above describes the general principle.

Analysis

Let's assume "ideal" hash functions (each key is mapped to a slot uniformly and independently). How can we upper bound the cost of the operations? Suppose we perform n operations. Let S be the set of elements that were ever inserted. We will assume $m > 2n$. We will consider the state of the hash table as if we had inserted all the elements in S.

The hash table A will be broken into "runs", where a run is a maximal contiguous interval of occupied cells. The key observation is the following: The cost of `insert(x)`, `find(x)`, and `delete(x)` are all upper-bounded by the length of the run that contains the slot $h(x)$. Let's call this run $R(x)$.

Our goal is to analyze the expected length of this run. What is $E[|R(x)|]$?

Lemma 1. *If $m > 2en$, under the ideal hashing assumption, then for any x, $E[|R(x)|] = O(1)$.*

This lemma implies that the total expected cost for n operations is $O(n)$.

Proof. Let R denote the run $R(x)$, and let $h(x) = i$. The expected length of the run is:

$$E[|R|] = \sum_{l=1}^{n} l \cdot P[|R| = l]$$

Consider an interval I of length l that contains the slot i. There are l such possible intervals. Let's fix one such interval, say I_j. For the run containing i to be exactly this interval I_j, two conditions must be met:

1. All l slots in I_j must be occupied by elements from S.
2. The two slots bordering I_j must be empty.

By symmetry, the probability that the run R is equal to any specific valid interval of length l is the same. Thus, $P[|R| = l]$ is related to $l \cdot P[R = I_j]$ for a fixed I_j.

Let's find an upper bound on the probability that a specific interval I_j of length l contains the run. For this to happen, at least l elements from S must hash into I_j. The probability of this is:

$$P[\text{at least } l \text{ items of } S \text{ hash to } I_j] \leq \binom{n}{l} \left(\frac{l}{m}\right)^l$$

Using the inequality $\binom{n}{k} \leq (\frac{en}{k})^k$, we get:

$$\leq \left(\frac{en}{l}\right)^l \left(\frac{l}{m}\right)^l = \left(\frac{en}{m}\right)^l$$

Given our assumption that $m \geq 2en$, we have $\frac{en}{m} \leq \frac{1}{2}$. Therefore, the probability is at most $(\frac{1}{2})^l$. Now we can bound the expected run length.

$$E[|R|] \leq \sum_{l=1}^{n} l \cdot \left(l \cdot \frac{1}{2^l}\right) = \sum_{l=1}^{n} \frac{l^2}{2^l}$$

This sum $\sum_{l=1}^{\infty} l^2 x^l$ converges for $|x| < 1$. For $x = 1/2$, the sum is a constant. Thus, $E[|R|] = O(1)$. □

Analysis with 5-Universal Hashing

The above analysis assumed ideal hashing. It turns out that a similar result can be obtained with a weaker assumption: 5-universal hashing.

Lemma 2. *Suppose H is a 5-strongly universal hash family from $[u] \to [m]$ and $m \geq 8n$. Then the expected cost of each of the first n operations is $O(1)$.*

To prove this, we need a concentration inequality for 4-wise independent random variables, which generalizes Chebyshev's inequality.

Lemma 3 (Concentration Inequality)**.** *Suppose $X_1, \ldots, X_n \in \{0, 1\}$ are 4-wise independent random variables. Let $X = \sum_{i=1}^{n} X_i$ and $\mu = E[X]$. Then for any $\beta > 0$:*

$$P[X \geq \mu + \beta] \leq \frac{\mu + 3\mu^2}{\beta^4}$$

Proof of $O(1)$ cost using 5-universal hashing. Let's assume the concentration lemma. We want to bound $E[|R|]$. We can write:

$$E[|R|] = \sum_{l=1}^{\infty} P[|R| \geq l] \leq \sum_{k=1}^{\lceil \log n \rceil} 2^k P[|R| > 2^{k-1}]$$

Let's bound the probability $P[|R| > 2^{k-1}]$. Let $h(x) = i$. Consider an interval I_k of length 2^{k+1} centered at i. If the run $R(x)$ has length greater than 2^{k-1}, then at least 2^{k-1} elements must have hashed into an interval of size at most 2^{k+1} around i.

Let X_a be the indicator random variable that element $a \in S \setminus \{x\}$ hashes into I_k. Let $X = \sum_{a \in S \setminus \{x\}} X_a$. Since our hash family is 5-strongly universal, conditioning on $h(x) = i$ leaves the hash values of other elements 4-wise independent. Thus, the variables X_a are 4-wise independent.

The expected value of X is:

$$\mu = E[X] = \sum_{a \in S \setminus \{x\}} E[X_a] = \sum_{a \in S \setminus \{x\}} P[h(a) \in I_k] \leq n \cdot \frac{|I_k|}{m} = n \cdot \frac{2^{k+1}}{m}$$

With $m \geq 8n$, we have:

$$\mu \leq n \cdot \frac{2^{k+1}}{8n} = \frac{2^{k+1}}{8} = 2^{k-2}$$

The event $|R| > 2^{k-1}$ implies that $X \geq 2^{k-1}$. We use our concentration lemma to bound this probability. Let $\beta = 2^{k-1} - \mu \geq 2^{k-1} - 2^{k-2} = 2^{k-2}$.

$$P[X \geq 2^{k-1}] = P[X \geq \mu + (2^{k-1} - \mu)] \leq P[X \geq \mu + 2^{k-2}]$$

Applying the lemma with $\beta = 2^{k-2}$ and $\mu \leq 2^{k-2}$:

$$\leq \frac{\mu + 3\mu^2}{\beta^4} \leq \frac{2^{k-2} + 3(2^{k-2})^2}{(2^{k-2})^4} = \frac{1 + 3 \cdot 2^{k-2}}{(2^{k-2})^3} \leq \frac{4 \cdot 2^{k-2}}{(2^{k-2})^3} = \frac{4}{(2^{k-2})^2}$$

Now we plug this back into the sum for $E[|R|]$:

$$E[|R|] \leq \sum_{k=1}^{\lceil \log n \rceil} 2^k \cdot P[|R| > 2^{k-1}] \leq \sum_{k=1}^{\infty} 2^k \cdot \frac{4}{(2^{k-2})^2} = \sum_{k=1}^{\infty} 2^k \cdot \frac{4}{2^{2k-4}} = \sum_{k=1}^{\infty} \frac{64 \cdot 2^k}{2^{2k}} = \sum_{k=1}^{\infty} \frac{64}{2^k}$$

This is a convergent geometric series, so $E[|R|] = O(1)$. □

Proof of Concentration Lemma. We want to prove $P[X \geq \mu + \beta] \leq \frac{\mu + 3\mu^2}{\beta^4}$. Using Markov's inequality on the non-negative random variable $(X - \mu)^4$:

$$P[X - \mu \geq \beta] = P[(X - \mu)^4 \geq \beta^4] \leq \frac{E[(X - \mu)^4]}{\beta^4}$$

Our task is to bound $E[(X-\mu)^4]$. Let $p_i = E[X_i]$, so $\mu = \sum p_i$. Let $Y_i = X_i - p_i$. Note that $E[Y_i] = 0$. Then $X - \mu = \sum Y_i$.

$$E[(X-\mu)^4] = E\left[\left(\sum_{i=1}^{n} Y_i\right)^4\right] = E\left[\sum_{i,j,k,l \in [n]} Y_i Y_j Y_k Y_l\right]$$

By linearity of expectation, this is $\sum_{i,j,k,l} E[Y_i Y_j Y_k Y_l]$. Because the variables are 4-wise independent and $E[Y_i] = 0$, any term where an index appears only once will be zero. For example, $E[Y_i Y_j^3] = E[Y_i]E[Y_j^3] = 0$ for $i \neq j$. The only non-zero terms are of the form $E[Y_i^4]$ and $E[Y_i^2 Y_j^2]$ for $i \neq j$. The expansion of $(\sum Y_i)^4$ gives:

- Terms of type Y_i^4: There are n such terms. The term is $\sum_i E[Y_i^4]$.
- Terms of type $Y_i^2 Y_j^2$: There are $\binom{n}{2}$ pairs $\{i, j\}$, and the coefficient for each is $\binom{4}{2} = 6$. The term is $6 \sum_{i<j} E[Y_i^2]E[Y_j^2]$.

So, $E[(X-\mu)^4] = \sum_{i=1}^{n} E[Y_i^4] + 6 \sum_{i<j} E[Y_i^2]E[Y_j^2]$. Let's bound these expectations. Since $X_i \in \{0, 1\}$, $X_i^k = X_i$ for $k \geq 1$. $E[Y_i^2] = E[(X_i - p_i)^2] = E[X_i^2 - 2p_i X_i + p_i^2] = p_i - 2p_i^2 + p_i^2 = p_i - p_i^2 = p_i(1-p_i) \leq p_i$. $E[Y_i^4] = E[(X_i - p_i)^4] = p_i(1-p_i)^4 + (1-p_i)(-p_i)^4 = p_i(1-p_i)^4 + (1-p_i)p_i^4$. Since $(1-p_i) \leq 1$ and $p_i \leq 1$, we have $E[Y_i^4] \leq p_i(1-p_i) + p_i^2(1-p_i) \leq p_i$. So, $\sum_i E[Y_i^4] \leq \sum_i p_i = \mu$. And $6 \sum_{i<j} E[Y_i^2]E[Y_j^2] = 6 \sum_{i<j} p_i(1-p_i)p_j(1-p_j) \leq 6 \sum_{i<j} p_i p_j$. We know that $2 \sum_{i<j} p_i p_j = (\sum_i p_i)^2 - \sum_i p_i^2 \leq (\sum_i p_i)^2 = \mu^2$. So, $6 \sum_{i<j} p_i p_j \leq 3\mu^2$. Putting it all together:

$$E[(X-\mu)^4] \leq \mu + 3\mu^2$$

Finally, substituting this back into the Markov inequality expression gives the desired result:

$$P[X - \mu \geq \beta] \leq \frac{\mu + 3\mu^2}{\beta^4}$$

□