CS 457, Fall 2019

Drexel University, Department of Computer Science

Lecture 9

Hash Tables

- Use a hash function h
  - Function h maps U to slots of hash table
  - Given key k, compute the slot $h(k)$
  - This reduces the required table size

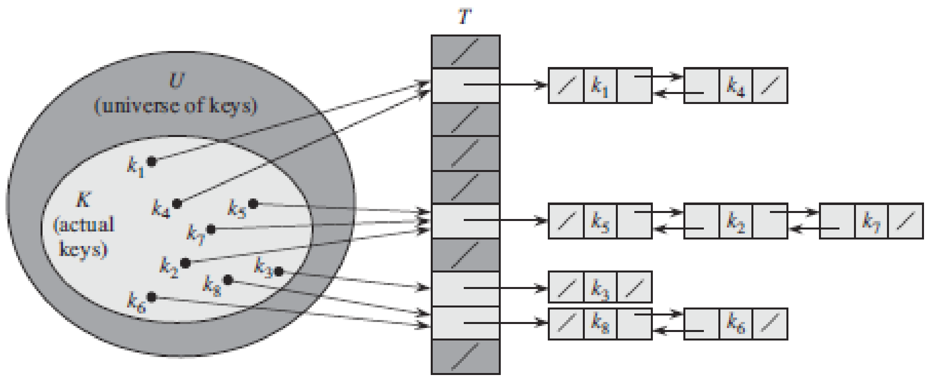
CHAINED-HASH-INSERT(T, x)
insert x at the head of list $T[h(x.key)]$

CHAINED-HASH-SEARCH(T, k)
search for an element with key k in list $T[h(k)]$

CHAINED-HASH-DELETE(T, x)
delete x from the list $T[h(x.key)]$

- But what if we get a collision?
  - Two distinct keys could be mapped to the same slot
  - How can we try to avoid this?
  - The function needs to be deterministic

- We can address that using chaining
  - Place colliding keys to same linked list
  - How does this affect the running time?



Hash Tables (Running Time)

- Given a hash table with m slots that stores n elements:
  - Worst case running time for searching is $\Theta(n)$ plus time to compute hash function
  - This is no better than the time achieved by a single linked list…
  - Simple uniform hashing: any element is equally likely to hash into any of the slots
- What about the average-case running time for search?
  - Let n/m be the load factor α for hash table T
  - Let n_j be the length of the list $T[j]$ for $j \in \{0, 1, \dots, m-1\}$
  - The expected value of n_j for uniform hashing is $\mathrm{E}[n_j] = \alpha$
  - Assume that computing the hash value $h(k)$ takes $O(1)$ time

Theorem 11.1

In a hash table in which collisions are resolved by chaining, an $\boxed{\text{unsuccessful}}$ search takes average-case time $\Theta(1+\alpha)$, under the assumption of simple uniform hashing.

Hash Tables (Running Time)

- Given a hash table with m slots that stores n elements:
  - Worst case running time for searching is $\Theta(n)$ plus time to compute hash function
  - This is no better than the time achieved by a single linked list…
  - Simple uniform hashing: any element is equally likely to hash into any of the slots
- What about the average-case running time for search?
  - Let n/m

Number of examined elements is: $X = \sum_{j=1}^{m} \frac{1}{m}\left(1 + n_j\right)$

So, $\mathrm{E}[X] = \sum_{j=1}^{m} \frac{1}{m}\left(1 + \mathrm{E}[n_j]\right) = \sum_{j=1}^{m} \frac{1}{m}\left(1 + \frac{n}{m}\right) = 1 + \frac{n}{m}$

Theorem 11.1

In a hash table in which collisions are resolved by chaining, an $\boxed{\text{unsuccessful}}$ search takes average-case time $\Theta(1+\alpha)$, under the assumption of simple uniform hashing.

Hash Tables (Running Time)

- Given a hash table with m slots that stores n elements:
  - Worst case running time for searching is $\Theta(n)$ plus time to compute hash function
  - This is no better than the time achieved by a single linked list…
  - Simple uniform hashing: any element is equally likely to hash into any of the slots
- What about the average-case running time for search?
  - Let n/m be the load factor α for hash table T
  - Let n_j be the length of the list $T[j]$ for $j \in \{0, 1, \ldots, m-1\}$
  - The expected value of n_j for uniform hashing is $\mathrm{E}[n_j] = \alpha$
  - Assume that computing the hash value $h(k)$ takes $O(1)$ time

Theorem 11.2
In a hash table in which collisions are resolved by chaining, a $\boxed{\text{successful}}$ search takes average-case time $\Theta(1+\alpha)$, under the assumption of simple uniform hashing.

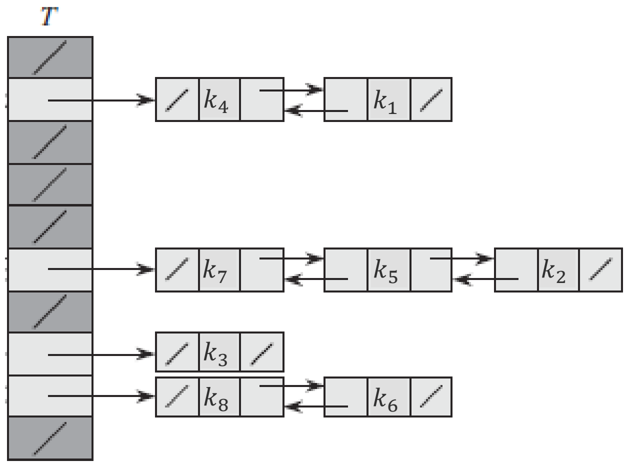
Hash Tables (Running Time)

- For keys k_i and k_j we define indicator variable $X_{ij} = \mathbb{I}\{h(k_i) = h(k_j)\}$
- For simple uniform … $\} = 1/m$
- Assume that … to be any of the n elements
- Expected num… essful search is:

For simplicity, this assumes that k_i is the key of the i-th element to be added to the hash table!

$$\mathrm{E}\left[\frac{1}{n}\sum_{i=1}^{n}\left(1+\sum_{j=i+1}^{n} X_{ij}\right)\right]$$

Verify this for the following instance:



Hash Tables (Running Time)

- For keys k_i and k_j we define indicator variable $X_{ij} = \mathbb{I}\{h(k_i) = h(k_j)\}$
- For simple uniform hashing, we get $\Pr\{h(k_i) = h(k_j)\} = 1/m$
- Assume that element being searched for is equally likely to be any of the n elements
- Expected number of elements examined in a successful search is:

$$
\begin{aligned}
\mathrm{E}\left[\frac{1}{n}\sum_{i=1}^{n}\left(1 + \sum_{j=i+1}^{n} X_{ij}\right)\right]
&= \frac{1}{n}\sum_{i=1}^{n}\left(1 + \sum_{j=i+1}^{n} \mathrm{E}\left[X_{ij}\right]\right) \quad \text{(by linearity of expectation)} \\
&= \frac{1}{n}\sum_{i=1}^{n}\left(1 + \sum_{j=i+1}^{n} \frac{1}{m}\right) \\
&= 1 + \frac{1}{nm}\sum_{i=1}^{n}(n - i)
\end{aligned}
$$

Hash Tables (Running Time)

$$
\begin{aligned}
\mathrm{E}\left[\frac{1}{n}\sum_{i=1}^{n}\left(1+\sum_{j=i+1}^{n} X_{ij}\right)\right] \\
&= \frac{1}{n}\sum_{i=1}^{n}\left(1+\sum_{j=i+1}^{n} \mathrm{E}\left[X_{ij}\right]\right) \quad \text{(by linearity of expectation)} \\
&= \frac{1}{n}\sum_{i=1}^{n}\left(1+\sum_{j=i+1}^{n} \frac{1}{m}\right) \\
&= 1+\frac{1}{nm}\sum_{i=1}^{n}(n-i) \\
&= 1+\frac{1}{nm}\left(\sum_{i=1}^{n} n-\sum_{i=1}^{n} i\right) \\
&= 1+\frac{1}{nm}\left(n^2-\frac{n(n+1)}{2}\right) \quad \text{(by equation (A.1))} \\
&= 1+\frac{n-1}{2m} \\
&= 1+\frac{\alpha}{2}-\frac{\alpha}{2n}\,.
\end{aligned}
$$

Today's Lecture

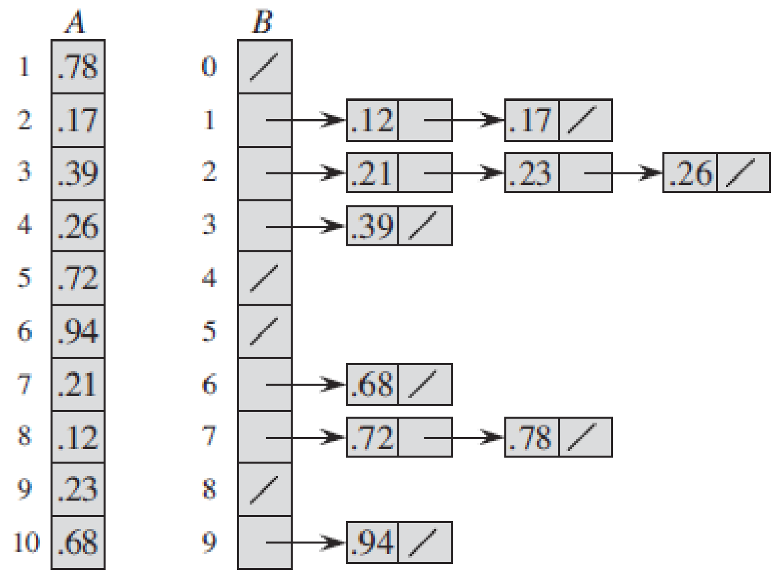
- More probabilistic analysis and randomized algorithms
  - Bucket Sort
  - Binary trees

Sorting in Linear Time

- Assume that the input is drawn from uniform distribution from interval $[0, 1)$
- Can we use this information to get a faster algorithm, in the worst case sense?
- Can we use this information to get a faster algorithm, in the average case sense?

BUCKET-SORT(A)

let $B[0\,.\,.\boxed{n-1}]$ be a new array
$n = A.length$
for $i = 0$ **to** $n - 1$
    make $B[i]$ an empty list
for $i = 1$ **to** n
    insert $A[i]$ into list $B[\lfloor nA[i] \rfloor]$
for $i = 0$ **to** $n - 1$
    sort list $B[i]$ with insertion sort
concatenate the lists $B[0], B[1], \ldots, B[n-1]$ together in order



Bucket Sort (Running Time)

- Let n_i be the number of elements in $B[i]$ (random variable)
- What is the running time, $T(n)$ of bucket sort as a function of n_i?
  - $T(n) = \Theta(n) + \sum_{i=0}^{n-1} O(n_i^2)$
- If we let $X_{ij} = \mathbb{I}\{A[j] \text{ falls in bucket } i\}$, then $n_i = \sum_{j=1}^{n} X_{ij}$

$$
\begin{aligned}
\mathrm{E}[T(n)] &= \mathrm{E}\left[\Theta(n) + \sum_{i=0}^{n-1} O(n_i^2)\right] \\
&= \Theta(n) + \sum_{i=0}^{n-1} \mathrm{E}\left[O(n_i^2)\right] \quad \text{(by linearity of expectation)} \\
&= \Theta(n) + \sum_{i=0}^{n-1} O\left(\mathrm{E}\left[n_i^2\right]\right) \quad \text{(by equation (C.22))} .
\end{aligned}
$$

It suffices to show that $\sum_{i=0}^{n-1} O(\mathrm{E}[n_i^2])$ is $\Theta(n)$

Bucket Sort (Running Time)

- So, it suffices to show that $\sum_{i=0}^{n-1} O(\mathrm{E}[n_i^2])$ is $\Theta(n)$
- What is the value of $\mathrm{E}[n_i^2]$?

$$
\begin{aligned}
\mathrm{E}\left[n_i^2\right] &= \mathrm{E}\left[\left(\sum_{j=1}^{n} X_{ij}\right)^2\right] \\
&= \mathrm{E}\left[\sum_{j=1}^{n}\sum_{k=1}^{n} X_{ij}X_{ik}\right] \\
&= \mathrm{E}\left[\sum_{j=1}^{n} X_{ij}^2 + \sum_{1\le j\le n}\ \sum_{\substack{1\le k\le n \\ k\ne j}} X_{ij}X_{ik}\right] \\
&= \sum_{j=1}^{n} \mathrm{E}\left[X_{ij}^2\right] + \sum_{1\le j\le n}\ \sum_{\substack{1\le k\le n \\ k\ne j}} \mathrm{E}\left[X_{ij}X_{ik}\right] ,
\end{aligned}
$$

Bucket Sort (Running Time)

- So, it suffices to show that $\sum_{i=0}^{n-1} O(\mathrm{E}[n_i^2])$ is $\Theta(n)$
- What is the value of $\mathrm{E}[n_i^2]$?

$$
\begin{aligned}
\mathrm{E}\left[n_i^2\right] &= \mathrm{E}\left[\left(\sum_{j=1}^{n} X_{ij}\right)^2\right] \\
&= \mathrm{E}\left[\sum_{j=1}^{n}\sum_{k=1}^{n} X_{ij}X_{ik}\right] \\
&= \mathrm{E}\left[\sum_{j=1}^{n} X_{ij}^2 + \sum_{1\le j\le n}\ \sum_{\substack{1\le k\le n \\ k\ne j}} X_{ij}X_{ik}\right] \\
&= \sum_{j=1}^{n}\mathrm{E}\left[X_{ij}^2\right] + \sum_{1\le j\le n}\ \sum_{\substack{1\le k\le n \\ k\ne j}} \mathrm{E}\left[X_{ij}X_{ik}\right] ,
\end{aligned}
$$

$$
\begin{aligned}
\mathrm{E}\left[X_{ij}^2\right] &= 1^2\cdot\frac{1}{n} + 0^2\cdot\left(1-\frac{1}{n}\right) \\
&= \frac{1}{n} .
\end{aligned}
$$

$$
\begin{aligned}
\mathrm{E}\left[X_{ij}X_{ik}\right] &= \mathrm{E}\left[X_{ij}\right]\mathrm{E}\left[X_{ik}\right] \\
&= \frac{1}{n}\cdot\frac{1}{n} \\
&= \frac{1}{n^2} .
\end{aligned}
$$

Simple Indicator Variables Example

Randomized-Loops (n)

$c = 0$
for $x = 1$ **to** n
    $k = \text{Random}(1, n)$ // Random number from 1 to n
    for $y = 1$ **to** k
        $c = c + 1$
return c

- What is the expected running time of Randomized-Loops(n)?
  - Let $X_{ij} = \mathbb{I}\{\text{The value of } k \text{ in the } i\text{–th outer loop iteration is equal to } j\}$
  - Then we wish to compute expected value of $X = \sum_{i=1}^{n} \sum_{j=1}^{n} j\, X_{ij}$
  - Taking the expectation, we get

$$\mathrm{E}[X] = \mathrm{E}\left[\sum_{i=1}^{n} \sum_{j=1}^{n} j\, X_{ij}\right] = \sum_{i=1}^{n} \sum_{j=1}^{n} j\, \mathrm{E}[X_{ij}] = \sum_{i=1}^{n} \sum_{j=1}^{n} \frac{j}{n} = \sum_{i=1}^{n} \frac{n+1}{2} = \frac{n(n+1)}{2}$$