Honors Algorithms
G22.3520-001 Fall 2006

Lecture 1
Instructor: Victor Shoup, WWH 511
Office hours: Mon/Wed 5–6pm

A “hybrid” course:
- Algorithms (text = CLRS)
- Automata and Complexity (text = Sipser)
Grading:

- problem sets 40%
- final exam 60%
  - doubles as CS Dept. Algorithms Exam

Course web page: http://www.cs.nyu.edu/courses/fall06/G22.3520-001/index.html

Course mailing list: See web page – be sure to subscribe!
Hashing

Reading: CLRS, Ch. 11, appendix C.1–4; CINTA, §§6.1–7, §6.10.

- $\mathcal{U}$ – a set of “data items”
- $T[0 \ldots m - 1]$ – a table for storing data, indices are called slots, buckets, or bins
- $h: \mathcal{U} \rightarrow \{0, \ldots, m - 1\}$ – a “hash function” maps data items to slots
- A collision is a pair $(a, b)$ such that $a \neq b$ but $h(a) = h(b)$
Resolving collisions by chaining
Dictionary Operations:

- **insert(\(a\))**: insert \(a\) in the linked list \(T[\(h(a)\)]\)
- **search(\(a\))**: search for \(a\) in \(T[\(h(a)\)]\)
- **delete(\(a\))**: search for and delete \(a\) in \(T[\(h(a)\)]\)

Running times:

- insert – \(O(1)\)
- search, delete – \(O(n)\) (worst case)

Worst case occurs when all items hash to the same slot.

Better: choose a *random* hash function hopefully — no “pile ups”
Universal Hashing  [Carter & Wegman, 1975]

- $\mathcal{K}$ – a finite, non-empty set of **hash keys**
- $\mathcal{H} = \{ h_k \}_{k \in \mathcal{K}}$ – a **family** of hash functions $h_k : \mathcal{U} \rightarrow \{0, \ldots, m - 1\}$, indexed by $\mathcal{K}$

**Def’n:** $\mathcal{H}$ is called **universal** if for all $a, b \in \mathcal{U}$ with $a \neq b$,

$$\left| \{ k \in \mathcal{K} : h_k(a) = h_k(b) \} \right| \leq \frac{|\mathcal{K}|}{m}.$$

**Probabilistic interpretation:** if $K$ is a random variable, uniformly distributed over $\mathcal{K}$, then

$$\Pr[h_K(a) = h_K(b)] \leq \frac{1}{m}.$$
Using Universal Hash Functions
Assume items $a_1, \ldots, a_n$ are stored in table
Let $\alpha := n/m = \text{“load factor”}$
Assume $K$ is uniformly distributed over $\mathcal{K}$
For $i = 1, \ldots, n$, define
$$S_i := \# \text{ of items in slot } h_K(a_i)$$
That is, $S_i$ is the number of items in the same slot as $a_i$
The values $K, S_1, \ldots, S_n$ are random variables.
For each $i = 1, \ldots, n$, we wish to bound
$$E[S_i] := \text{the expected value of } S_i.$$
Claim: $E[S_i] \leq \alpha + 1$ for each $i = 1, \ldots, n$.

Proof: for $i, j = 1, \ldots, n$, define

$$C_{ij} := \begin{cases} 
1 & \text{if } h_K(a_i) = h_K(a_j) \\
0 & \text{otherwise}
\end{cases}$$

Each $C_{ij}$ is called an indicator variable.

For $i \neq j$, we have

$$\Pr[C_{ij}] \leq 1/m \quad \text{(def'n of universal hashing)}$$

$$E[C_{ij}] = 1 \cdot \Pr[C_{ij} = 1] + 0 \cdot \Pr[C_{ij} = 0]$$

$$(\text{def'n of expectation})$$

$$\leq 1/m$$

For each $i$: $\Pr[C_{ii}] = 1$ and $E[C_{ii}] = 1$
By definition, we have

\[ S_i = \sum_{j=1}^{n} C_{ij} \]

By linearity of expectation, we have

\[
E[S_i] = \sum_{j=1}^{n} E[C_{ij}]
= E[C_{ii}] + \sum_{j \neq i} E[C_{ij}]
\leq 1 + \frac{n - 1}{m}
\leq \alpha + 1 \quad \text{QED}
\]
interpretation:

- for each $i$, the expected # of items in $a_i$'s slot (including $a_i$ itself) is $\leq \alpha + 1$
- the expected time to perform a single dictionary operation is $O(\alpha + 1)$
- by linearity of expectation, expected time to perform $n$ dictionary operations is $O(n(\alpha + 1))$

**special case:** $\alpha = O(1)$ (i.e., $n = O(m)$)

- expected time per operation is $O(1)$
Maximum Load: another performance measure.

Suppose hash table contains items $a_1, \ldots, a_n$, and that $K$ is uniform over $\mathcal{K}$

For $s = 0, \ldots, m - 1$, define

$$L_s := \# \text{ of } a_i \text{'s that hash to slot } s \text{ under } h_K$$

Set $M := \max\{L_s : s = 0, \ldots, m - 1\}$

We want to bound $E[M]$, assuming universal hashing

**Fact:** $E[M]^2 \leq E[M^2]$

**Fact:** $M^2 \leq V := \sum_{s=0}^{m-1} L_s^2$

**Claim:** $E[V] \leq n^2/m + n$
Proof of claim: Define indicator variables

\[ I_{i,s} := \begin{cases} 1 & \text{if } h_{K}(a_i) = s \\ 0 & \text{otherwise} \end{cases} \]

We have

\[ V = \sum_{s=0}^{m-1} L_s^2 = \sum_{s=0}^{m-1} \left( \sum_{i=1}^{n} I_{i,s} \right)^2 \]

\[ = \sum_{s} \left( \sum_{i} I_{i,s} \right) \left( \sum_{j} I_{j,s} \right) \]

\[ = \sum_{i,j} \sum_{s} I_{i,s} I_{j,s} = \sum_{i,j} C_{ij} \]
So we have

\[ V = \sum_{i,j} C_{ij} \]

and by linearity of expectation, we have

\[ E[V] = \sum_{i,j} E[C_{ij}] \]

\[ = \sum_i E[C_{ii}] + \sum_{i \neq j} E[C_{ij}] \]

\[ \leq n + n(n - 1)/m \]

\[ \leq n^2/m + n \]

QED
Corollary: \( E[M] \leq \sqrt{n^2/m + n} \)

Special case: \( \alpha = O(1) \) (i.e., \( n = O(m) \))

\[
E[M] = O(\sqrt{m})
\]

- This bound is tight
- Counter-intuitive: it may be the case that \( E[L_s] = O(1) \) for each slot \( s \)

General fact: expected value of max may be much larger than max of expected values