# Lecture #11: Discrete Probability for Computer Science II

## What To Do Before the Lecture

1. Watch the videos for Lecture #11 — noting that they will probably be understandable if you play them at double speed. If you do not have time for this then look at the "Key Concepts" document that is found, immediately after the videos for this lecture, on the course web site, instead.
2. ***Print*** and read through the rest of this document and — if you have time — try to solve the problems!

## Problems To Be Solved

### A "Data Structures" Example: Expected Performance of Hashing

1. Consider the representation of a ***finite set***

   $$\{k_1, k_2, \ldots, k_n\}$$

   of $n$ values from a larger set, or "universe", $\mathcal{U}$. A ***hash table with chaining*** is a data structure — often discussed in a course like CPSC 331 — that can be used to represent this kind of set.

   For a positive integer $m$ — which we will call the ***table size*** — a ***hash function*** (for the universe $U$ and table size $m$) is a total function

   $$h : \mathcal{U} \to \{0, 1, \ldots, m-1\} \tag{1}$$

   mapping each element $k \in U$ to an integer $h(k)$, such that $0 \leq h(k) \leq m-1$. This is used to access, and modify, the data structure being described.

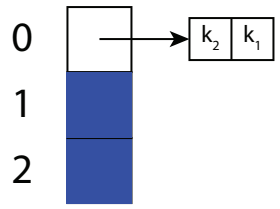
   In general, a hash table representing the set $\{k_1, k_2, \ldots, k_n\}$, with table size $m$, is an ***array*** $T$ with length $m$. For each integer $i$ such that $0 \leq i \leq m-1$, $T[i]$ is a reference to a *list* of the values

   $$S_i = \{k_j \mid 1 \leq j \leq n \text{ and } h(k_j) = i\}, \tag{2}$$

where $h : \mathcal{U} \to \{0, 1, 2, \ldots, m-1\}$ is the hash function used to construct this hash table.

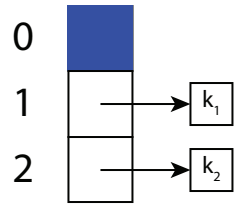
The hash table is constructed by starting with an "empty" table — and array $T$ such that $T[i]$ is a reference to an empty list, for each integer $i$ such that $0 \leq i \leq m-1$. Then, for $1 \leq j \leq n$ (in increasing order), key $k_j$ is inserted at the *front* of the linked list at position $T[h(k_j)]$.

For example, if $m = 3$ and $n = 2$, then the hash table corresponding to the outcome $(0, 0)$ (where $h(k_1) = h(k_2) = 0$) is as follows.




In this and later pictures, a cell shown in blue stores a reference to an empty list. Thus — using the notation given above, $S_0 = \{k_1, k_2\}$ and $S_1 = S_2 = \emptyset$.

On the other hand, if $m = 3$ and $n = 2$, then the hash table corresponding to the outcome $(1, 2)$ (where $h(k_1) = 1$ and $h(k_2) = 2$) is as follows.




In this case $S_0 = \emptyset$, $S_1 = \{k_1\}$, and $S_2 = \{k_2\}$.

Since the hash table (used to represent the above set $\{k_1, k_2, \ldots, k_n\}$) only depends on the "hash values" $h(k_1), h(k_2), \ldots, h(k_n)$, each ***outcome*** can be represented as a sequence

$$\vec{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_n) \tag{3}$$

where $\alpha_i = h(k_i) \in \{0, 1, 2, \ldots, m-1\}$ for each integer $i$ such that $1 \leq i \leq n$. That is, the ***sample space***, to be used here, is the set

$$\Omega_{n,m} = \{(\alpha_1, \alpha_2, \ldots, \alpha_n) \mid \alpha_i \in \mathbb{Z} \text{ and } 0 \leq \alpha_i \leq m-1 \text{ for every integer } i \text{ such that } 1 \leq i \leq n\}. \tag{4}$$

Algorithms to insert values into hash tables, delete values from them, and search for elements in them, all start by applying the hash function to determine which of

$$T[0], T_1], \ldots, T[m-1]$$

should be a reference to the link list that would store this value, if was found in the hash table. Then algorithm then continues with a search in the linked list (and an update to it, if the operation requires this) — so that, for a hash table as above, the number of steps needed perform a search or update operation is at most linear in

$$\max(|S_0|, |S_1|, |S_2|, \ldots, |S_{m-1}|). \tag{5}$$

Since all the elements that are stored could be in the same list — so that $|S_i| = n$, the size of the set being represented, for some integer $i$ such that $0 \leq i \leq m-1$, the *worst-case* performance of operations in a hash table is poor. Performance can be much better — especially if if the value shown at line (5), above, is significantly less than $n$.

With that noted, let us consider the following random variables:

- For $0 \leq i \leq m-1$, $X_i : \Omega \to \mathbb{R}$ represents the size, $|S_i|$, of the set of keys $k \in \{k_1, k_2, \ldots, k_n\}$ such that $h(k) = i$. Thus, for an outcome $\vec{\alpha} \in \Omega_{n,m}$ as shown at line (3), above, $X_i(\vec{\alpha})$ is the number of integers $j$ such that $1 \leq j \leq n$ and $\alpha_j = i$.
- For $0 \leq i \leq m-1$ and $1 \leq j \leq n$, $X_{i,j}$ is equal to $1$ when $h(k_j) = i$ and $X_{i,j}$ is equal to $0$, otherwise. Thus, for an outcome $\vec{\alpha} \in \Omega_{n,m}$ as shown at line (3), above,

$$X_{i,j}(\vec{\alpha}) = \begin{cases} 1 & \text{if } \alpha_j = i, \\ 0 & \text{otherwise.} \end{cases}$$

(a) How is $X_i$ related to $X_{i,1}, X_{i,2}, \ldots, X_{i,n}$? What does this imply about the relationship between the expected value of $X_i$ and the values of $X_{i,1}, X_{i,2}, \ldots, X_{i,n}$, with respect to some probability distribution?

In order to perform an "average-case analysis" it is necessary to make an assumption about the likelihood that various hash tables are used. Let us assume that the hash values are uniformly and independently assigned to keys — that we will be using the ***uniform probability distribution***, that is, the function

$$\mathsf{P} : \Omega_{n,m} \to \mathbb{R}$$

such that

$$\mathsf{P}(\sigma) = \frac{1}{|\Omega_{n,m}|} = m^{-n}$$

for every outcome $\sigma \in \Omega_{n,m}$.

(b) Compute the ***expected value*** of the random variable $X_{i,j}$ with respect to this probability distribution (where $i$ and $j$ are integers such that $0 \leq i \leq m-1$ and $1 \leq j \leq n$).

(c) Use this compute the ***expected value*** of $X_i$.

Henceforth, let $k$ be an integer such that $2 \leq k \leq m$. For the rest of this question, let us consider bounds on the probability that $X_i \geq \frac{kn}{m}$ that can be obtained using the results that were given in the preparatory material for this lecture.

(d) State the best bound that you can on the probability that $X_i \geq \frac{kn}{m}$ that can be obtained using ***Markov's Inequality***.

(e) Now compute the ***variance*** of $X_{i,j}$, for integers $i$ and $j$ such that $0 \leq i \leq m-1$ and $1 \leq i \leq n$.

(f) State — as specifically as you can (and relating this to *this problem* instead of simply repeating a definition) what you would need to show, in order to argue that the random variables $X_{i,1}, X_{i,2}, \ldots, X_{i,n}$ are ***pairwise independent*** with respect to the probability distribution that is being used.

(g) State — as specifically as you can (and relating this to *this problem* instead of simply repeating a definition) what you would need to show, in order to argue that the random variables $X_{i,1}, X_{i,2}, \ldots, X_{i,n}$ are ***mutually independent*** with respect to the probability distribution that is being used.

For the rest of this question you may assume that the random variables $X_{i,1}, X_{i,2}, \ldots, X_{i,n}$ are, indeed, ***mutually independent*** with respect to this probability distribution, for every integer $i$ such that $1 \leq i \leq n$.

(h) Compute the ***variance*** of the random variable $X_i$ — listing all properties of the random variables $X_i$ and $X_{i,1}, X_{i,2}, \ldots, X_{i,n}$ that you used to do this.

(i) State the best bound on the probability that $X_i \geq \frac{kn}{m}$ that you can using both versions of ***Chebyshev's Inequality***, and using ***Cantelli's Inequality.***

(j) Can the ***Chernoff Bound*** be applied, here, to bound the probability that $X_i \geq \frac{kn}{m}$? If it cannot, then explain why that is the case. Otherwise, use this bound to bound this probability as well.

(k) As noted above, the number of steps needed to perform a search or update on this hash table is at most linear in

$$\max(|S_0|, |S_1|, |S_2|, \ldots, |S_{m-1}|).$$

With that noted, let

$$X = \max(X_0, X_1, X_2, \ldots, X_{m-1}).$$

Let $\varepsilon$ be a real number such that $\varepsilon > 0$. Use the above to find a value $N$ — which will depend on $n$, $m$ and $\varepsilon$ — such that

$$\mathsf{P}(X \geq N) \leq \varepsilon.$$

2. Recall that — as stated in the preparatory material — ***Chebyshev's Inequality*** is as follows.

   **Theorem 7** (Chebyshev's Inequality)**.** *Let $\Omega$ be a* **finite** *sample space with probability distribution $P : \Omega \to \mathbb{R}$, let $X$ be a random variable, and let $a \in \mathbb{R}$ such that $a > 0$. Then*

$$\mathsf{P}(|X| \geq a) \leq \frac{\mathsf{E}[X^2]}{a^2}.$$

   *and*

$$\mathsf{P}(|X - \mathsf{E}[X]| \geq a) \leq \frac{\mathsf{var}(X)}{a^2}.$$

   Give a proof of the this theorem.