8 Probability in Hashing

A popular method for storing a collection of items to support fast look-up is hashing them into a table. Trouble starts when we attempt to store more than one item in the same slot. The efficiency of all hashing algorithms depends on how often this happens. The problem itself is of course universal as things and beings are often organized by putting them into boxes, drawers, cages, taxis, etc.

Birthday paradox. We begin with an instructive question about birthdays. Consider a group of $n$ people. Each person claims one particular day of the year as her birthday. For simplicity, we assume that nobody claims February 29 and we talk about years consisting of $k = 365$ days only. Assume also that each day is equally likely for each person. In other words,

$$P(\text{person } i \text{ is born on day } j) = \frac{1}{k},$$

for all people $i$ and all days $j$. Collecting the birthdays of the $n$ people, we get a multiset of $n$ days during the year. We are interested in the event, $A$, that at least two people have the same birthday. Its probability is one minus the probability that the $n$ birthdays are distinct, that is,

$$P(\bar{A}) = 1 - P(A) = 1 - \frac{k}{k} \cdot \frac{k-1}{k} \cdots \frac{k-n+1}{k} = 1 - \frac{1}{(k-n)!k^n}.$$

The probability of $A$ surpasses one half when $n$ exceeds 21, which is perhaps surprisingly early. See Figure 11 for a display how the probability grows with increasing $n$.

![Figure 11: The probability that at least two people in a group of $n$ share the same birthday.](image)

Hashing. The basic mechanism in hashing is the same as in the assignment of birthdays. We have $n$ items and map each to one of $k$ slots. We assume the $n$ choices of slots are independent. A collision is the event that an item is mapped to a slot that already stores an item. There is a variety of strategies we can use to resolve a collision, such as moving the colliding item to the first empty slot nearby, or by randomly assigning a new slot, or by allowing more than one item in a slot. However, this will not be our concern in this section. We are more interested in getting an understanding of the probability of collisions and empty slots. More precisely, we are interested in the following quantities:

1. the expected number of items mapping to same slot;
2. the expected number of empty slots;
3. the expected number of collisions;
4. the expected number of items needed to fill all $k$ slots.

Different hashing algorithms use different mechanisms for resolving collisions. The above quantities have a lot to say about the relative merits of these algorithms.

**Items per slot.** Since all slots are the same and none is more preferred than any other, we might as well determine the expected number of items that are mapped to slot 1. Consider the corresponding indicator random variable,

$$X_i = \begin{cases} 1 & \text{if item } i \text{ is mapped to slot 1;} \\ 0 & \text{otherwise.} \end{cases}$$

The number of items mapped to slot 1 is therefore $X = X_1 + X_2 + \ldots + X_n$. The expected value of $X_i$ is $\frac{1}{k}$, for each $i$. Hence, the expected number of items mapped to slot 1 is

$$E(X) = \sum_{i=1}^{n} E(X_i) = \frac{n}{k}.$$

But this is obvious in any case. As mentioned earlier, the expected number of items is the same for every slot. Writing $Y_j$ for the number of items mapped to slot $j$, we have $Y = \sum_{j=1}^{k} Y_j = n$. Similarly,

$$E(Y) = \sum_{j=1}^{k} E(Y_j) = n.$$

Since the expectations are the same for all slots, we therefore have $E(Y_j) = \frac{n}{k}$, for each $j$. 

21
Empty slots. The probability that slot $j$ remains empty after mapping all $n$ items is $(1 - \frac{j}{k})^n$. Defining

\[ X_j = \begin{cases} 
1 & \text{if slot } j \text{ remains empty;} \\
0 & \text{otherwise}, 
\end{cases} \]

we thus get $E(X_j) = (1 - \frac{1}{k})^n$. The number of empty slots is $X = X_1 + X_2 + \ldots + X_k$. Hence, the expected number of empty slots is

\[ E(X) = \sum_{j=1}^{k} E(X_j) = k \left( 1 - \frac{1}{k} \right)^n. \]

For $k = n$, we have $\lim_{n \to \infty} (1 - \frac{1}{n})^n = e^{-1} = 0.367 \ldots$. In this case, we can expect about a third of the slots to remain empty.

Collisions. The number of collisions can be determined from the number of empty slots. Writing $X$ for the number of empty slots, as before, we have $k - X$ items hashed without collision and therefore a total of $n - k + X$ collisions. Writing $Z$ for the number of collisions, we thus get

\[ E(Z) = n - k + E(X) = n - k + k \left( 1 - \frac{1}{k} \right)^n. \]

For $k = n$, we get $\lim_{n \to \infty} n(1 - \frac{1}{n})^n = e^{-1} = 0.367$. In words, about a third of the items cause a collision.

Filling all slots. How many items do we need to map to the $k$ slots until they store at least one item each? For obvious reasons, this question is sometimes referred to as the coupons collector problem. The crucial idea here is to define $X_j$ equal to the number of items it takes to go from $j = 1$ to $j$ filled slots. Filling the $j$-th slot is an infinite Bernoulli process with success probability equal to $p = \frac{k - j + 1}{k - j}$. Last section, we learned that the expected number of trials until the first success is $\frac{1}{p}$. Hence, $E(X_j) = \frac{k}{k - j + 1}$. The number of items needed to fill all slots is $X = X_1 + X_2 + \ldots + X_k$. The expected number is therefore

\[ E(X) = \sum_{j=1}^{k} E(X_j) = k \sum_{j=1}^{k} \frac{k}{k - j + 1} = k \sum_{j=1}^{k} \frac{1}{j}. \]

The result of this sum is known as the $k$-th harmonic number: $H_k = 1 + \frac{1}{2} + \frac{1}{3} + \ldots + \frac{1}{k}$. We use $\int_{x=1}^{k} \frac{dx}{x} = \ln x$ to show that the $k$-th harmonic number is roughly the natural logarithm of $k$. More precisely, we get

\[ H_k \leq 1 + \int_{x=1}^{k} \frac{dx}{x} = 1 + \ln k, \]

\[ H_k \geq \int_{x=1}^{k+1} \frac{dx}{x} = \ln(k + 1). \]

To summarize, the expected number of items we need to fill every slot in a table of size $k$ is about $k \ln k$. This is an important result about sampling: if you study a population of $n$ individuals by uniform sampling then you need an expected number of about $n \ln n$ samples to reach each individual.