DEWEZZ Lu

V00897211

CSC226: Lab 9

1 0 or 1 Hash Tables

Suppose we have a table with m slots to store values. Values are hashed into the table randomly, and each slot of the table contains a 1 if a value has been hashed to that index, and 0 otherwise. The probability that a particular value gets tossed to a specific slot is $\frac{1}{m}$. The following table illustrates an example of such a hash table.

indices	0	1	2	3	4	$\cdots$	$m-2$	$m-1$
values	1	0	0	1	0	$\cdots$	1	0

A hash function $h_1(k)$ determines the slot to which value k will belong. Suppose that n distinct values have been added to the table, and the values all went to unique locations (by chance). When a value k is hashed into the table, we do not store k itself, but a 1 to indicate that k has been inserted into position $h_1(k)$. For this hash table, there are no "colisions" as we've seen in class since if we insert an element l at position $h_1(l)$, and we find that the value at position $h_1(l)$ already equals 1, we do nothing.

We would like to know if a particular value v has been inserted into the table. The hash value of v is $h_1(v)$. If the value at index $h_1(v)$ is 0, then v is not in the table. If the value at index $h_1(v)$ is 1, then one of two scenarios occured:

1. v was added to the table. We call this a **true positive**.
2. v was not added to the table, but another value had been hashed to the same location as $h_1(v)$. We call this a **False Positive**.

We want to find the probability of getting a false positive. If we know that there are n values already in our table of size m, then the probability of getting a 1 in the table by chance is the total number of 1's in the table divided by the total number of slots in the table. That is, the probability of a false positive is $\frac{n}{m}$.

Is it possible to lower the chances of getting a false positive? Consider if we add another hash table with its own hash function $h_2(k)$. When we add k under this scheme, it gets added to both hash tables using $h_1(k)$ and $h_2(k)$ for their respective tables. How will this change the probability of getting false positives? When looking to see if some value is in our tables, we say that it is **only if** if there is both a 1 at $h_1(k)$ and a 1 and $h_2(k)$. The probability that k gives a false positive in either table (we know from above) is $\frac{n}{m}$. These are independent events, which means that the probability of both of them happening is equal to multiplying the probabilities of each event happening. Therefore, the probability of getting a false positive with two tables and two hash functions is $(\frac{n}{m})^2$.

It is possible
since $(\frac{n}{m})^2$ both 1 at $h_1(k)$ and a1 and $h_2(k)$. The
probability that k gives a false positive.

2 Exercises

For the lab submission, please submit your solutions to the following two questions in a single PDF file.

1. What is the probability of false positive when we have η hash tables each with their own hash function?

2. Let's remove the assumption that the first n values go into unique spots. Does the probability of false positives go up or down from this change, and why? (Note: the number of tables used does not affect this answer)

1. It is only possible there is both a_1 at hit with n hash table and a_2 in the dim (k). The probability that be gives a false positive is in either functions

2. It will definitely go up. ~~The~~ Since the number of table does not affect the answer. The probability ~~it~~ does not suffer from how the death would be applied to the thing. It will only go up when the sample size went up