

Miscellaneous notes in mathematics

Abiyaz Chowdhury

October 29, 2018

1 Functions

Let A be an array of elements having n entries. For some positive integer k , we wish to find all entries of A that appear more than n/k times in A . We setup a table of size $k - 1$, where each table contains an element and the counter for that element. Iterate through the array. If an element is in the table, increment its counter. If it is not in the table, but there is space in the table, add it to the table with a counter value of 1. If no space is in the table, decrement every counter in the table.

Claim 1 If an element x appears more than n/k times in A , it will be stored in the table when the algorithm terminates.

Claim 2 For any element in A , its counter value in the table (0 if it is not in the table) is no greater than its actual count in the array, and is no less than its actual count minus $\lfloor n/k \rfloor$.

Theorem 1.1 ■ Let Σ be the sum of all counter values at any moment. Σ is always nonnegative (starts out at 0), and either increases by 1 or decreases by $k - 1$ when a new element is seen.

Theorem 1.2 ■ Σ can not be decremented more than n/k times.

Theorem 1.3 ■ For each element x with frequency $f(x) > n/k$, define $c(x)$ to be the counter value if x is in the table, and $-\Sigma$ otherwise. To prove claim 1, it suffices to show that at the end of the algorithm, $c(x) > 0$.

Theorem 1.4 ■ To achieve the above, we need to guarantee that $c(x)$ increases frequently enough times and decreases infrequently enough so that its final value is positive. We need to find when it could increase or decrease.

Theorem 1.5 ■ If x is in the table and an x is encountered, $c(x)$ increases by 1. If x is in the table, and something other than x is encountered, $c(x)$ either decreases by 1 or doesn't change depending on whether the table is full.

Theorem 1.6 ■ If x is not in the table, and an x is encountered, $c(x)$ must increase. If the table had room before the x was encountered, $c(x)$ will increase from $-\Sigma$ to 1. If the table had no room before the x was encountered, every counter in the table is decremented. The new value of $c(x) = -(\Sigma - (k - 1)) = -\Sigma + (k - 1)$. If x is not in the table, and some other element is encountered that is in the table, $c(x)$ will decrease by 1. If x is not in the table, and some other element is encountered not in the table, $c(x)$ will decrease by 1 if there is room for it, and increase by $k - 1$ if there is no room for it.

Theorem 1.7 ■ Combining the above results, $c(x)$ must increase whenever an x is encountered. So $c(x)$ increases by more than n/k over the course of the algorithm.

1. At any time t (i.e. at iteration t), for any element x , define $g_t(x) = f_t(x) - \text{decr}_t(x)$ where $f_t(x)$ is the number of times x is seen up to time t and $\text{decr}_t(x)$ is the number of times a full table decrement is performed using an x .
2. $g_t(x) \geq 0$ for all t .
3. $g_n(x) > 0$ if x is a heavy hitter.
4. $g_t(x) > 0$ if and only if x is in the table at time t . Furthermore, if x is in the table at time t , then $c_t(x) = g_t(x)$.

We are done if we prove the above. We restrict our analysis to $k = 2$.

5. ■ $g(x)$ can only increment if an x is seen while the table either contains x or has room for x .
6. ■ If $g(x)$ is ever incremented, by the end of that step, x must be in the table.
7. ■ If $g_0(x) = 0$.
8. ■ $g(x)$ can decrement only if x is already in the table.
9. ■ Let's try an induction to prove claim 4. Initially, $g(x) = 0$ and x is not in the table. Suppose that claim 4 holds for all t less than i . Prove that it holds for $t = i$. When the i th element is encountered,