Treasure Hunting

Zheng Zangwei*

April 5, 2018

Problem 1

(a)Given an array A[0...n-1], to determine whether there is a value that occurs more than $\lfloor \frac{n}{k} \rfloor$ times in $\Theta(nlgk)$ time and $\Theta(k)$ extra space.
(b)Prove that the lower bound of this problem is $\Theta(nlgk)$.

1 Take n=2

Algorithm 1

Search from the beginning to end with a stack. If the stack is empty, then push the number else pop it. The element in stack is the element we want (If we are not sure whether there is, then judge this element again in O(n))

Obviously it is $\Theta(n)$ in time and $\Theta(1)$ in space. .
This algorithm is correct. Exchange the element with 1 and others with -1, each time we operate on stack can be viewed as adding 1 or -1 to a counter. At last, the counter will ber greater than 0.

2 k-simplified Multiset

Definition 1

k-simplified multiset: a multiset derived from a multiset M by repeating deleting k distinct elements from it untill no longer possible.

Theorem 2.1

If a value occurs more than $\lfloor \frac{n}{k} \rfloor$ times in M then it is in the k-simplified multiset.

*E-mail: zzw@smail.nju.edu.com Student ID: 171860658

Proof

Suppose all values are $a_1, a_2, ..., a_n$ in a descending order according to their appearing times and the first so occur more than $\lfloor \frac{n}{k} \rfloor$ times. Suppose the worst case that a_s appears only $\lfloor \frac{n}{k} \rfloor$ times. since k distinct values can delete a a_s then we need to prove:

$$\lfloor \frac{n - \lfloor \frac{n}{k} \rfloor}{k-1} \rfloor < \lfloor \frac{n}{k} \rfloor + 1$$

which turns out to be $\lfloor \frac{n}{k} \rfloor + k - 1 > \frac{n}{k}$ which is correct. □

By this proof, we can generalize an algorithm of $\theta(nk)$

Algorithm 2

(1)For every element x in the set

(1.1) If x is in Keepset then count[x]++ and put x in Keepset

(1.2) If the size of Keepset equals k then

(1.2.1) For every element y in Keepset

(1.2.1.1) count[y] -= 1

(1.2.1.2) If count[y] == 0 delete y from Keepset

(2) Check elements in Keepset by traversing the set again

(3) Output the satisfactory elements

3 Improved Data Structure

In Algorithm 2 the Keepset is a linear data structure. Each node contains two information: values V and times of appearance T. This time we use a balanced tree to help. With the tree, the three basic operations that we omit their proof here are 1. put in a number 2. search for a number 3. delete the maximal number from the heap are all $\Theta(logn)$.

Algorithm 3

(1)For every element x in the set

(1.1) If x is in Keepset then

(1.1.1) Search node that (node.V==x) in Keepset

(1.1.2) If (finded) node.T += 1

(1.1.3) Otherwise make node(V=x,T=1) and insert into Keepset.

(1.2) If the size of Keepset equals k then

(1.2.1) For every element y in Keepset

(1.2.1.1) count[y] -= 1

(1.2.1.2) If count[y] == 0 delete y from Keepset

(2) For every element x in the set

(2.1) Search node that (node.V==x)

(2.2) If (finded) node.V' += 1

(3) Output node.V for nodes in Keepset that node.V'>$\lfloor \frac{n}{k} \rfloor$ elements

Sentence (1.2.1.2) is not a basic function of a balanced tree but we can do it since we can view it and it's springs as another max heap and execute the basic operation 3. This algorithm costs $\Theta(nlogk)$ in time and $\Theta(k)$ extra space.

4 Another algorithm

Recalling that we have an algorithm to k-sort an array in nlog(k) times, we have this algorithm:

Algorithm 4

(1) k-sort the array.

(2) Judge first-k elements.

Since there are $\lfloor \frac{n}{k} \rfloor$ groups, thus the number we desire must occur at least once in every groups.

5 The Proof of the lower bound

We assume that this algorithm is based on comparison. We have the theorem:

Theorem 5.1

Every algorithm based on comparison can be analysed by determintric tree.

Definition 2

Let $r = \lfloor \frac{n}{k} \rfloor$. An r-list of n values such that each of the values 0,1,...,$\lfloor \frac{n}{r} \rfloor - 1$ occurs r times and value $\lfloor \frac{n}{r} \rfloor$ occurs (n mod r) times.

Lemma 1

The number of different r-lists of n values is bigger than $\frac{k}{e}^n$

Proof

(This proof is not that strict)

Denote the number of different r-lists as A.

$$\begin{aligned} A &= \frac{n!}{\lfloor \frac{n}{k} \rfloor^{\lfloor \frac{n}{r} \rfloor - 1} * (n\%r)!} \\ &= \frac{n!}{\frac{n}{k}^k} \\ (asymptotic) &= \frac{\sqrt{2\pi n}\frac{n}{e}^n}{(\sqrt{2\pi\frac{n}{k}} * \frac{n}{ke}^{\frac{n}{k})^k)} \\ &> \frac{k^n}{\sqrt{2\pi n * (1/k)}^k}(*) \end{aligned}$$

To prove $(*) > \frac{k}{e}^n$*, we only need to prove* $ln\pi + ln\frac{2n}{k} < \frac{2n}{k}$*. This is asymptoticly true. Thus we have proved the lemma in an asymptotic sense.* □

Theorem 5.2

Executing these r-lists on a dicision tree will follow different paths.

Proof

Notice that all r-lists will result in an output of false. As we follow a certein path in decision tree, we can get two sets of relationship, one denots equation and another inequation. Consider two different r-lists A, B follows the same paths, so they have the same sets of relationship. These relationship are enough to output false and we will prove that they can determine only one r-lists. We also denote the input to be $a_1a_2...a_n$

Link a_i, a_j *with an edge if they have been checked to be the same. Call a set of points that are connected with edges group. The number of group is less than k and the number of points in a group is less than* $\lfloor \frac{n}{k} \rfloor$*. Values among a group is same. For two groups, if their values are the same, they can merge together and there is a risk that the number will exceed* $\lfloor \frac{n}{k} \rfloor$*.*

If any two numbers in two groups respectively are checked to be different, then we add an edge from the small one to the bigger one. Thus we get a DAG. We will show that by these edges we can determine the only possible value of each group.

(CORE)Consider the value a_p *which is different is A and B. Since the DAG holds, the group which contains this value must be one-size and all nodes if exists connected to it must be smaller than the minimum of* a_p *now and before. This is similar for nodes*

connected from it. Thus by replacing a_p with the former values, we get a output of true from this path, which is a contradiction.

In this way, all values have been determined. Thus there is only one possibility. □

Remark

The first time I use equation and inequation as the node of the determining tree which tough two can share the same paths but we didn't use all information. Thus it cannot deduce the time we deal by comparison.

The core part of my past proof is listed as follows.

Disproof

(only consider rk=n)If a group contains r elements then it must be linked to all other groups so that this group has been determined. If only one values was divded into many groups since we have determined the other groups. Otherwise, there are two values been divded into several groups.

In the picture, s_i, t_i all denotes a group of certain elements. Then s_i must be checked

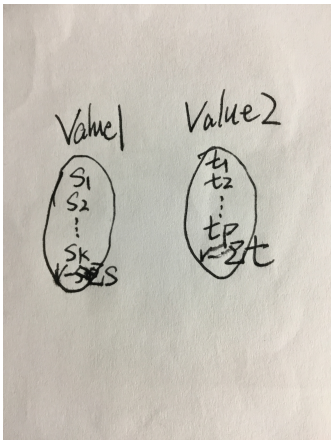

Figure 1: The last situcation

with all t for add all s together plus t will exceed $\lfloor n \rfloor k$ (check any two s will not change this situation because this only decrease the scale of the problem). This is the same to t. Thus we can tell them apart.

Hence we have at least $\frac{k}{e}^n$ different paths which means at least $\frac{k}{e}^n$ leaves. So we have the worst height of the tree h, $2^h > \frac{k}{e}^n$. Therefore $h = \Theta(nlogk)$. Thus we know the lower bound of this problem is $\Theta(nlgk)$.