

Good Problems

Ahmed Zaheer Dadarkar
tymefighter

June 30, 2020

- 1 All lengths to which a string can be compressed
- 2 Number Of Occurrences of Every Prefix
- 3 Small To Large

You are given a set $S = \{S_1, \dots, S_n\}$ consisting of n multisets each currently containing a single element. You have to perform $n - 1$ operations, in each operation you would be given i and j and you have to merge the sets S_i and S_j and replace them with $S_{\min(i,j)}$ and then provide the number of distinct elements in that set. Method is that when asked to merge two sets, place the elements of the smaller set into the larger one. Now, consider any element $x \in \bigcup_{i=1}^n S_i$, this element is added to a new set $O(\log n)$ times, the reason being that the size of the set this element is part of becomes atleast double of the previous one. Hence, after m merges, we have

$$2^m \leq n$$

So,

$$m = O(\log n)$$

Now, each add operation has $O(\log n)$ cost, hence for each element we have a total cost of $O(\log^2 n)$, So for all elements together we have time complexity

$$O(n \log^2 n)$$

3.1 DSU using size of component

The method is that each time we are asked to merge two components, we make the node having subtree with larger size as the parent of the other. Then at each such operation, we increase the size of the smaller tree by atleast a factor of 2.

Hence, since each op costs $O(\log n)$, and there are $O(\log n)$ such operations for **each** element, the total complexity is $O(n \log^2 n)$

3.2 A Tree Problem

You are given a tree rooted at 0 and a color on each node, you are then asked to compute the number of distinct colours in the subtree of each node.

The method is to find out for each node which child is heavy, by heavy we mean that it has the largest size of subtree compared to other children. Then we merge the sets of all other children into the heavy child.

An important thing to note is that we must not create a new set for the parent, instead we must keep the reference to the heavy child's set as the parent.

For analysis, assume the color at each node is unique, meaning no other node has it. Now, pick color c at any node (initially), then c is moved at most $O(\log n)$ times since the set containing c has its size doubled at each operation. Also, cost of each operation is $O(\log \min(n, |C|))$ where C is the set of all colors, and when we consider all elements, we have the following complexity

$$O(n(\log n)(\log \min(n, |C|))) = O(n \log^2 n)$$

4 Query for Number of Distinct elements in a Subarray

5 References

- <https://cp-algorithms.com/string/z-function.html>
- <https://cp-algorithms.com/string/prefix-function.html>
- <http://codingwithrajarshi.blogspot.com/2018/06/small-to-large.html>
- <https://stackoverflow.com/questions/39787455/is-it-possible-to-query-number-of-distinct-integers-in-a-range-in-olg-n>