Problem Description

Given a stream of positive integers $S = a_1, a_2, \cdots a_m$ with $max(S) \leq n$. Compute the *geometric mean* of S using $\mathcal{O}(polylog(n) + polylog(m))$ memory.

Problem Simplification

A very useful property of the geometric mean is that it is the arithmetic mean of S in log-scale. In other words it has a homomorphic relationship with the arithmetic mean. Mathematically expressed as:

$$\text{Geometric Mean}(a_1, a_2, \cdots a_m) = exp(\frac{1}{m}\sum_{i=1}^{m} ln(a_i))$$

This is important since we do not need to keep track of the product of the element in S down the stream, but rather we need to keep track of the sum of the natural log of the elements, which is much smaller.

Algorithm

Using the above simplification, an algorithm can be used to find the arithmetic sum of the elements of S and when the stream is finished (or at any point in the stream) the geometric mean can be calculated by using the EXP function of the sum divided by the number of elements seen in the stream.

Algorithm 1 Geometric Mean

Input: S

1: $sum \leftarrow 0$
2: $seen \leftarrow 0$
3: **for** $a \in S$ **do**
4: $\quad sum \leftarrow ln(a) + sum$
5: $\quad seen \leftarrow seen + 1$
6: **end for**
7: $res \leftarrow exp(\frac{sum}{seen})$
8: **return** res

Proof of Correctness

An invariant α in the code is that $seen = \#elments \wedge sum = \sum_{i=1}^{seen} ln(a_i)$ this is trivially true before the loop, then in the *for* loop the number of elements increase by one and so does the value of *seen* on line 5. The current value of *sum* is also increased by a which in this case is a_{seen}, in terms of the sequence $S = a_1, \cdots a_{seen}, \cdots a_m$. As a result α is true through the entire *for* loop. This means that after the *for* every element has been seen and thus $seen = m$ and hence the value of *sum* is $\sum_{i=1}^{m} ln(a_i)$. Putting this in the simplification earlier we get that

$$\text{Geometric Mean} = exp(\frac{sum}{seen})$$

Which is the value that is returned on line 8. It is also to note that all elements $a \in S$ are positive so $ln(a)$ will always be defined.

Space Complexity

Given that the stream a stream of positive integers $S = a_1, a_2, \cdots a_m$ with $max(S) \leq n$. The worst case for this algorithm is when every element in the stream is equal to n, i.e. the stream of m elements with the value n.
In this case the value of *sum* would be $m \cdot ln(n)$. This is in the order of $\mathcal{O}(log(m \cdot ln(n))) = \mathcal{O}(log(m) + log(ln(n)))$ space.
The value of *res* would trivially be $res \leq n$, since the geometric mean cannot be greater than the largest element in the stream, and thus use $\mathcal{O}(log(n))$ memory.
The value of *seen* will be equal to M and use $\mathcal{O}(log(m))$ memory.
In total the memory used will be thus be the sum of all these

$$\mathcal{O}(log(m) + log(ln(n)) + log(n) + log(m)) = \mathcal{O}(log(m) + log(n))$$

.