Sum of Geometric Progression Series

Kalpana Khutela
IIB2019019

Devang Bharti
IIB2019020

Hitka
IIB2019021

Abstract—**In this report we used divide and conquer approach to find the sum of infinite GP**

Keyword : Geometric Progression, time complexity, space complexity,.

I. Introduction

In this problem, we have to find the sum of geometric progression series

For example: if we have n=5 and x=2, and as per question type of series is x,x2,x3,x4.... and so on. so common ratio will be x, so in above case it will be 2+4+8+16+32=62

we have given a brief idea of our algorithm **part II**.

II. Algorithm Approach

1) The algorithm asks for an integer 'n' as an input. which is stated as number of terms in that GP

2) Now the algorithm asks for a "X" value which will be the first term of that series.

3) Now we used the an application of divide and conquer approach named binary search to solve this problem.

4) Call for the GeometricSequenceSum function with parameters as a(array) and n(size of array) which will give the answer

5) GeometricSequenceSum function is using a recursive approach to find the answer.

6) Base conditon:
Return 0 if size=0
else if size=1 then return a[0] element

7) Create two variables middle and rightSize.Initialize middle=size/2 and rightSize = size-middle.

8) Create two variables leftSum and rightSum.

9) Call for the GeometricSequenceSum function with parameters as a(array) and middle(size of array) which will return its computed sum in leftSum variable.

10) Call for the GeometricSequenceSum function with parameters as (a+ middle)(array passed from a+middle index) and rightSize(size of array) which will return its computed sum in rightSum variable.

11) After receiving values in leftSum and rightSum , these values are being added and final computed sum will be returned.

12) Sum of the geometric sequence will be the required output and after performing the above stated actions the requirement will be satisfied.

For Example -
Let input be n=5 and x=2

array will be 2,4,8,16,32
now middle=2
right size=3

for left sum1
middle =1
right =1
for left sum1-2
size=1
so it will return a[0], that is 2
for left1 right sum1
size =1
it will return a[1], that is 4
and in final step we will have a[0]+a[1], that is 2+4=6

for right sum1
middle=1 right =2 for this middle it again go in function and size will be 1 so it will return a[2]
which is 8
and for right =2
middle =1
right =1
for this middle it again go in function and give a[3] that is 16
and for right it again go with size 1 and which will return a[4], that is 32
it will add 16,32,8 and give 56
which again called recursively and give 6+56, that is 62

III. Pseudo Code

Function GeometricSequenceSum
Pass : int arr[ ], int size

$basecase$
if $size == 0$ **then**
return 0
else if $size == 1$ **then**
return a[0]
end if
$intmiddle = size/2$
$intrightSize = size - middle$
$intleftSum = GeometricSequenceSum(a, middle)$
$intrightSum = GeometricSequenceSum(a + middle, rightSize)$

IV. Time Complexity

The overall complexity of the question is O(n). Our algorithm uses divide and conquer to find the sum of GP series.

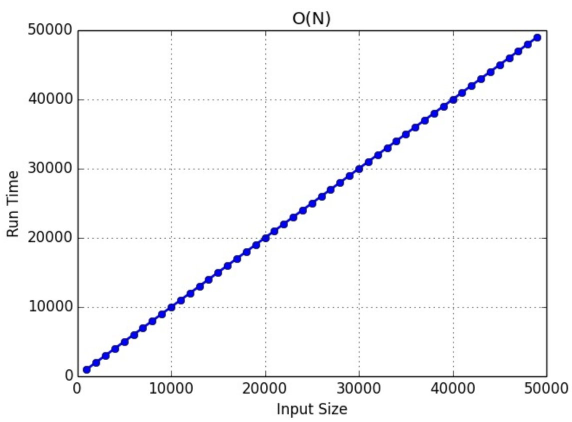

Fig. 1. Time complexity curve

V. Auxiliary Space Complexity

No extra space is used in this algorithm, so auxiliary space is constant.Only the input array is of size n. Space Complexity = Input Space+Auxiliary Space, which is equal to O(n).

Fig. 2. Auxiliary space complexity curve

VI. Conclusion

The above proposed algorithm efficiently gives the sum of the given series. The algorithm proposed is very efficient both space and time wise with O(n) time complexity.

VII. References

1) Geometric Progression series and sum

2) Divide and Conquer Algorithm — Introduction