Question: Find the big-Theta notation as a function of n of the following...

Find the big-Theta notation as a function of n of the following program:

```
q, s ← 1, 1
while s < n
   for k ← 1 to s
      p ← 1
      while p ≤ k
         p ← 2 * p
   q ← q + 1
   s ← q * q
```

Show transcribed image text

Expert Answer

Was this answer helpful? 2 0

General guidance

Show more

Step-by-step

FIRST STEP | ALL STEPS | ANSWER ONLY

Step 1 of 2

The best way to find the complexity (theta notation) of a code is to dry run the code for some initial values then try to generalize it.

Dry run of the given code:

Explanation

Please refer to solution in this step.

Step 2 of 2

DRY RUN:

```
1
2  ------------------- OUTER WHILE LOOP --------------------
3
4  n = a large value
5  initially q = 1 and s = 1
6
7  for the outer while loop, the value of q will increase by 1 in every iteration
8
9  the value of q in every iteration will be like = 1,2,3,4,5,6,7.......m ( m times )
10
11 the s will increase like s = 1*1 , 2*2, 3*3, 4*4, 5*5,.........,m*m
12
```

```
the loop wil

m ^ 2 = n

m = sqrt ( n )

So the running time of outer while loop is = sqrt ( n )


---------------------- INNER FOR LOOP ------------------------

the inner loop is running for k = 1 to s ( for every value of s )

iteration of this inner for loop will be like ( for every value of s )

1 to 1*1 ( 1^2 )

1 to 2*2 ( 2^2 )

1 to 3*3 ( 3^2 )

..
..
..

1 to sqrt(n) * sqrt(n)


= 1^2 + 2^2 + 3^2 + 4^2 +........+ sqrt(n)^2

using the formula 1^2 + 2^2 + .... + n^2 = n(n+1)(2n+1)/6

= sqrt(n)(sqrt(n) + 1)(2*sqrt(n) + 1) / 6

So the running time of inner for loop = sqrt(n)(sqrt(n) + 1)(2*sqrt(n) + 1) / 6


------------------- INNER WHILE LOOP ---------------------


the inner while loop is running for p = 1 to k ( for every value of k )

because in every iteration the value of p is increasing by a multiplying constant factor 2 so the time
complexity will be a logarithmic function

iteration of this inner while loop will be like ( for every value of k )

1 to log (1^2) == 1 to 2log1 ( because logm^n = nlogm )

1 to log (2^2) == 1 to 2log2

1 to log (3^2) == 1 to 2log3

..
..
..

1 to log (sqrt(n)^2) == 1 to 2log(sqrt(n))


= 2log1 + 2log2 + 2log3 +......+ 2log(sqrt(n))

= take 2 as common

= 2 ( log1 + log2 + log3 + .... log(sqrt(n)) )

using the property log x + log y = log x*y

= 2 ( log(1+2+3+..+sqrt(n))

so the running time will be :

= 2 ( log ((sqrt(n)(sqrt(n) +1 ))/2) ( because 1+2+3+4+...n = n(n+1)/2 )

-----------------------------------------------

Because all the loops are nested loop

So the running time complexity of this code is


sqrt ( n ) * ( sqrt(n)(sqrt(n) + 1)(2*sqrt(n) + 1) / 6 ) *  ( 2 ( log ((sqrt(n)(sqrt(n) +1 ))/2) )


Theta notation =
θ ( sqrt ( n ) * ( sqrt(n)(sqrt(n) + 1)(2*sqrt(n) + 1) / 6 ) *  ( 2 ( log ((sqrt(n)(sqrt(n) +1 ))/2) ) )
```

Explanation

Please refer to solution in this step.

As described in the
```
1) / 6 ) *  ( 2
```

Questions viewed by other students

Q: Consider Above Program Use big-Theta notation, the running time of the inner while loop, as a function of q is theta(q). a. Express the running time of the outer while loop as a sum. b. Express the total running time of the program, as a function (as simplified as possible) of n.

A: See answer

Q: Given that the number of iterations of the outer loop is approximately lg(n) for the following Program: Express, using big-Theta notation, the total running time of the program, as a function (as simplified as possible) of n.

A: See answer