

7:20 30
324018962

4 שאלה, 1 שאלה - 100%

a^n חזק של a ו- n מספר טבעי. $a \in \mathbb{R}$ (c)

הזמן $O(\log n)$ -

def multiply(a, n):

arrayA = powArray(a, n) // $[1, a, a^2, a^4, \dots, a^{\lceil \log n \rceil}]$

array2 = powArray(2, n) // $[1, 2, 4, 8, \dots, 2^{\lceil \log n \rceil}]$

answer = 1 // if $n=0$

counter = findClosestIdxFromBelow(n, array2) // $\lceil \log n \rceil$

$n = n - \text{array2}[\text{counter}]$

while ($n > 0$):

$n = n - \text{array2}[\text{counter}]$

answer = answer * arrayA[counter]

counter = findClosestIdxFromBelow(n, array2)

return answer.

def powArray(b, n):

c = array[size: $\lceil \log n \rceil$]

c[0] = 1

c[0] = b

for (i=2 to $\lceil \log n \rceil$)

c[i] = c[i-1] * c[i-1]

return c

def findClosestIdxFromBelow(n, array)

idx = 0

for (i=0 to array.size - 1)

if array[i] > n:

return idx

idx = i

def find x (n)		ⓐ
high = n		ה מספר מוצא בעצמו
low = 0		t = log n סמלית
x = $\frac{high + low}{2}$		אמצע המספרים high, low
while (x ² ≠ n):		אוקר O(t)
if x ² > n		חישוב x אוקר O(t) + O(t)
high = x		כפ. שהוכחנו דאלי 3
else		חישוב 2 מספרים בעלי סמלית
low = x		ב אחר אוקר O(t) (high + low)
x = $\frac{high + low}{2}$		אפסולת החילוק, שהיא פשוט
		Shift right העצת ב בים
return x		מינה, הנקרא O בים הנ אבוק

העצמת הבים הנ נמוך אחרי שהוצג

אוקרת ג מ O(t) (לעבור בים, קים, והעצ) וכן סה"כ O(t) + O(t) = O(t)
 חישוב הנקרא דגור ה while הנו O(t²) לחישוב x² (ב) שהוכחנו

דאליה 3, כן לחישוב מופה ל 2 מספרים בעלי t סמלית O(t) +
 בעצמות לשם ההשוואה ל x² - 1. n. לכן סה"כ O(t²)

חישוב הנקרא if אוקר O(t²) והעצת high = x או לחילופין
 low = x אוקרת O(t)

חישוב x = $\frac{high + low}{2}$ מוצא אוקרת כמחר O(t).

מספרים אר אולא ה - while כל הוצר log n בעצת (חישוב הנקרא),
 לומר t בעצת (log n = t).

לכן סה"כ בעצמות דאליהם :

$$O(t) + O(t) + O(t) + t \cdot (O(t^2) + O(t^2) + O(t) + O(t)) \Rightarrow O(t^3)$$

high	↑	low	↑	חישוב x	↑	חישוב while	↑	חישוב if	↑	חישוב x	↑	חישוב x
העצת		העצת		חישוב		חישוב while		חישוב if		חישוב		חישוב

לכן האלגוריתם ב זמן פולינומילי באופן הקרוב