Princekumar Diyora

65657055З

Assign:- 1

(1) $T(n) = 2T(n-1)+1$ for $n \geq 2$ $T(1)=1$

↓

recursive function

$T(n) = 2^n - 1$ for $n \geq 1$

Prove:- $T(K+1) = 2^{K+1} - 1$

Proof by Induction:-

Base case:-

for $n=2$

L.H.S $T(2) = 2T(2-1)+1$
$= 2T(1)+1$
$= 2(1)+1$
$\boxed{T(2) = 3}$

L.H.S = R.H.S

→ By definition, for $n=2$ given recursive function and explicit form of that is true.

R.H.S $T(2) = 2^2 - 1$
$= 4 - 1$
$\boxed{T(2) = 3}$

Induction Hypothesis:-

Suppose, for some $n=k$, where $k \geq 1$ that
$T(K) = 2^K - 1$

Induction Step:-

Then, for $n=k+1$, recursive function is

$T(K+1) = 2T(K+\cancel{1}-\cancel{1})+1$ (∵ recursive definition)
$= 2T(K)+1$
$= 2[2^K - 1]+1$ (∵ Induction Hypothesis)
$= 2^1 \cdot 2^K - 2 + 1$
$\boxed{T(K+1) = 2^{K+1} - 1}$

→ Since, $n = k+1$, $T(k+1) = 2^{k+1} - 1$

→ By Induction, the recursive function
$T(n) = 2T(n-1) + 1$ for $n \geq 2$ and $T(1) = 1$
has the explicit form $T(n) = 2^n - 1$ for $n \geq 1$.

2 $T(n) = T(n-1) + \log(n)$
↓ recursive function
for $n \geq 2$ & $T(1) = 0$
$T(n) = \log(n!)$ $n \geq 1$ → explicit form
$T(k+1) = \log[(k+1)!]$

Proof By Induction:

Base case:-

for $n = 2$

L.H.S $T(2) = T(2-1) + \log(2)$
$= T(1) + \log 2$
$= 0 + \log 2$
$\boxed{T(2) = \log 2}$

R.H.S $T(2) = \log(2!)$
$= \log(2 \times 1)$
$\boxed{T(2) = \log 2}$

L.H.S = R.H.S

→ By definition, for $n = 2$ given recursive function and its explicit form is True.

Induction Hypothesis:

suppose, for some $n=k$, $k\geq 1$ explicit form is $T(k) = \log(k!)$

Induction step:

Suppose, for some $n=k+1$, recursive function is

$$T(k+1) = T(k+\cancel{1}-\cancel{1}) + \log(k+1) \quad (\because \text{recursive definition})$$

$$= T(k) + \log(k+1)$$

$$= \log(k!) + \log(k+1) \quad (\because \text{Induction Hypothesis})$$

$$= \log(k!\,(k+1)) \quad (\because \text{using log rules for addition to multiplication})$$

$$= \log(\underbrace{(k+1) \times k!}_{(k+1)!}) \quad \text{we know that } (k+1)\,k! = (k+1)!$$

$$\boxed{T(k+1) = \log((k+1)!)}$$

Since, $n=k+1$, $\boxed{T(k+1) = \log[(k+1)!]}$

→ By Induction, the recursive function $T(n) = T(n-1) + \log(n)$ for $n\geq 2$, $T(1)=0$ has the explicit form $T(n) = \log(n!)$ for $n\geq 1$.

$T(2) = 2T(\frac{2}{2}) + 1 = 2 \times 2 + 1 = 5$
$T(4) = 2T(2) + 1 = 5 \times 2 + 1 = 11$

3

(9) $T(n) = 2T(\frac{n}{2}) + 1 \quad T(1) = 2$ ← recursive function

- what is explicit formula by recursive expansion?

→ for, simpler analysis let's assume $n = 2^k$, for some $k \geq 1$

$T(2^k) = 2T(\frac{2^k}{2}) + 1$ → This recursive expansion for $i = 1$;

$$\boxed{T(2^k) = 2T(2^{k-1}) + 1}$$

→ The recursive expansion for $i = 2$;

$$= 2[2T(2^{k-1-1}) + 1] + 1$$

$$= 2[2T(2^{k-2}) + 1] + 1$$

$$= 4T(2^{k-2}) + 2 + 1$$

→ The recursive expansion for $i = 3$;

$$= 4[2T(2^{k-1-2}) + 1] + 2 + 1$$

$$= 4[2T(2^{k-3}) + 1] + 2 + 1$$

$$= 8T(2^{k-3}) + 4 + 2 + 1$$

→ The recursive expansion for $i=4$

$$= 8\left[2T(2^{k-4})+1\right]+4+2+1$$

$$= 16\,T(2^{k-4})+8+4+2+1$$

→ Now, let's do the recursive expansion for ith term

$$= 2^i\,T(2^{k-i})+(2^i-1)$$

→ How, we got the ith term by analyzing sequence from 1st term to 4th term.

→ stop, recursive expansion when

$$k-i=1$$

$$\boxed{i=k-1}$$

→ so, $T(2^k) = 2^{k-1}\,T(2^{k-k+1}) + (2^{k-1}-1)$

$$= 2^{k-1}\,T(2^1) + 2^{k-1}-1$$

$$= 2^{k-1}\times 5 + 2^{k-1}-1 \quad \left(\because T(2)=2T\left(\tfrac{2}{2}\right)+1 = 2\times 2+1,\ T(2)=5\right)$$

$$= 2^{k-1}[5+1]-1 \quad (\because 6 = 3\times 2)$$

$$= 2^{k-1}\times 2\times 3-1$$

$$= 2^{k-1+1}\times 3-1$$

$$\boxed{T(2^k) = 3\,2^k-1}$$

→ now, we know that $n = 2^K$,

$$\boxed{T(n) = 3n - 1}$$

→ this is the explicit formula by recursive expansion.

3 b $T(n) = 2T\left(\frac{n}{2}\right) + 1$ $T(n) = 3n - 1$

recursive function explicit formula

$T(1) = 2$

Prove $T(K+1) = 3(K+1) - 1$
$= 3K + 3 - 1$
$= 3K + 2$

Proof By Induction:-

for, $n = 2$

L.H.S $T(2) = 2T\left(\frac{2}{2}\right) + 1$
$= 2\,T(1) + 1$ $(\because T(1) = 2)$
$= 2 \times 2 + 1$

$$\boxed{T(2) = 5}$$

R.H.S $T(2) = 3 \times 2 - 1$
$= 6 - 1$

$$\boxed{T(2) = 5}$$

→ L.H.S = R.H.S

→ By definition for n=2 given recursive function and its explicit form is true.

Induction Hypothesis:

Suppose, for some $n=k$, $k \geq 1$ explicit form is $T(k) = 3k-1$.

$= 2k + k - 1$

Induction step:

Suppose, for some $n=k+1$, recursive function is

$$T(k+1) = 2T\left(\frac{k+1}{2}\right) + 1$$

let's put $\frac{k+1}{2}$ into Induction Hypothesis

$$T\left(\frac{k+1}{2}\right) = 2\left(\frac{k+1}{2}\right) + \left(\frac{k+1}{2}\right) - 1$$

$$= \frac{2k + \cancel{2} + k + 1 - \cancel{2}}{2}$$

$$\boxed{T\left(\frac{k+1}{2}\right) = \frac{3k+1}{2}}$$ → Put this Value in our Induction step

$$T(k+1) = 2T\left(\frac{k+1}{2}\right) + 1$$

$$= \cancel{2}\left[\frac{3k+1}{\cancel{2}}\right] + 1$$

$$\boxed{T(k+1) = 3k+2}$$

→ since, $n = k+1$

$$T(k+1) = 3(k+1) - 1$$

$$= 3k + 3 - 1$$

$$\boxed{T(k+1) = 3k+2}$$

→ By, Induction recursive function $T(n) = 2T\left(\frac{n}{2}\right) + 1$ with $T(1) = 2$ has explicit formula $T(n) = 3n - 1$.

(4) The function which return the subset sizes in UnionFind Class.

→ I think, we have to make a function and inside that function we need to loop through the list of elements and we have to check that current index is equal to parent of the subset that we are looking for, and if it matches then we have to keep track of how many times it will happen. When the index doesn't match to the parent of the subset it will come outside from the loop and we have to return the last result which will be the size of the subset.

```
function name (Passing the integer Value)
{
        // we have to take a integer variable

      for ( go through all the elements)
      {
            if(m-Parent index will match
                 to the m-Parent subset)
            {
                 // we have to move on and on.
            }
            //else it will go outside of
             loop
      }
    //return the size of subset
```