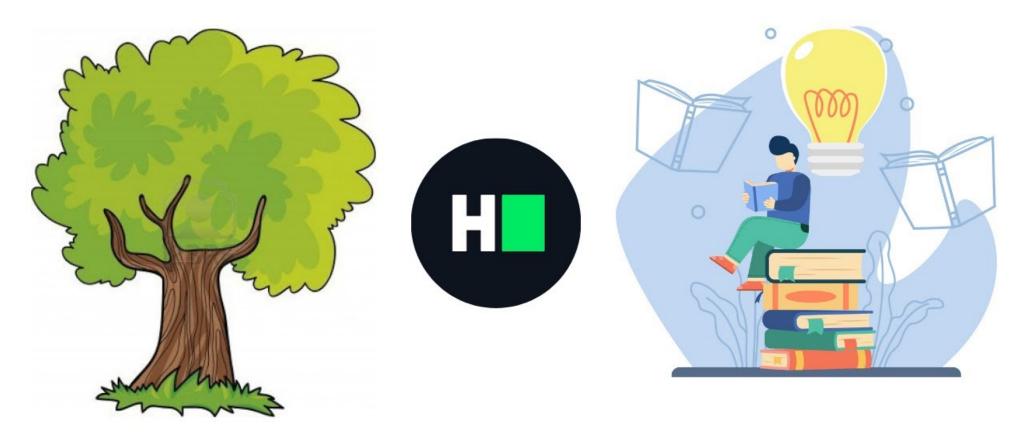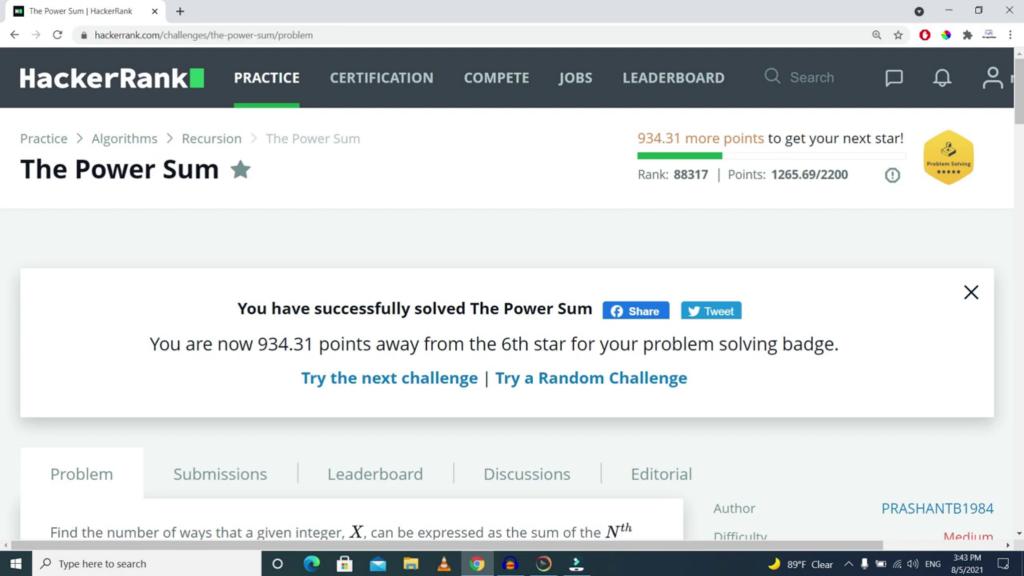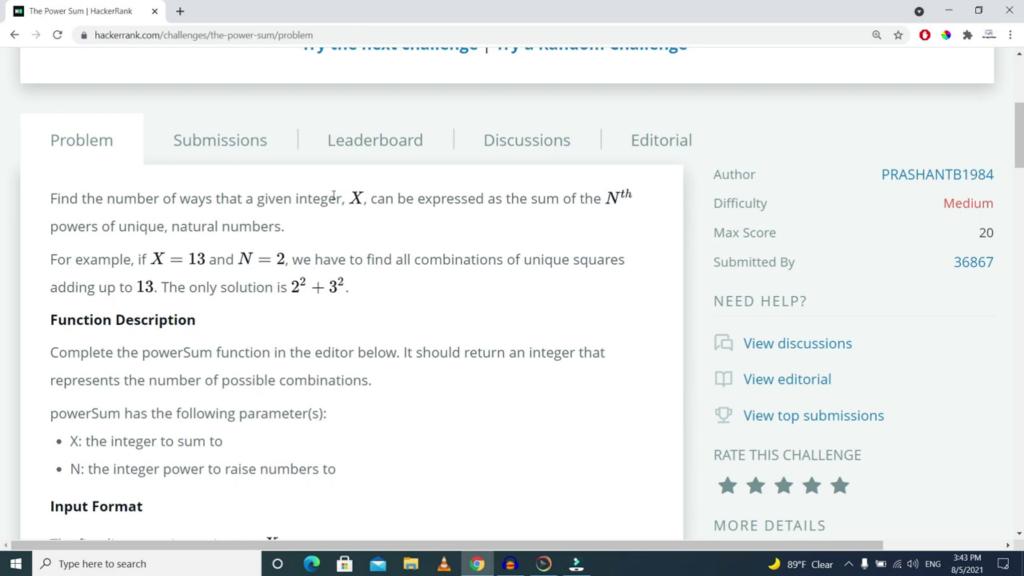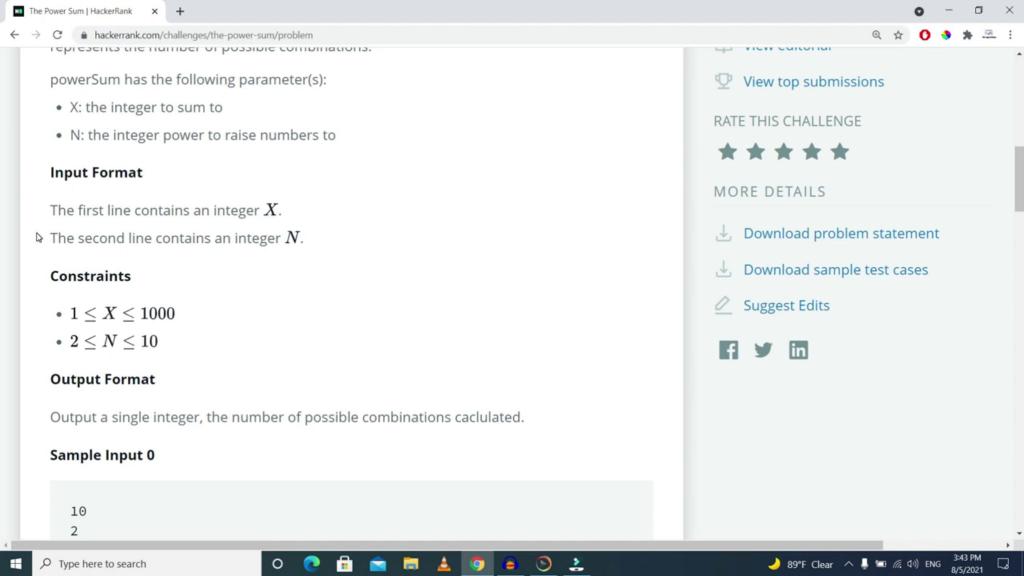# Explaining the Recursion tree through a problem

prior knowledge

1-what's recursion (vidoe on my channel)

2-recursive examples (also on my channel)

HackerRank **PRACTICE**   **CERTIFICATION**   **COMPETE**   **JOBS**   **LEADERBOARD**   🔍 Search

Practice > Algorithms > Recursion > The Power Sum

The Power Sum ⭐

934.31 more points to get your next star!

Rank: **88317** | Points: **1265.69/2200** ⓘ

Problem Solving
★★★★★

✕

You have successfully solved The Power Sum 🅕 **Share**   🐦 **Tweet**

You are now 934.31 points away from the 6th star for your problem solving badge.

Try the next challenge | **Try a Random Challenge**

| Problem | Submissions | Leaderboard | Discussions | Editorial |

Author   PRASHANTB1984

Find the number of ways that a given integer, X, can be expressed as the sum of the Nth Difficulty   Medium

Problem   Submissions | Leaderboard | Discussions | Editorial

Author	PRASHANTB1984
Difficulty	Medium
Max Score	20
Submitted By	36867

Find the number of ways that a given integer, X, can be expressed as the sum of the Nth powers of unique, natural numbers.

For example, if $X = 13$ and $N = 2$, we have to find all combinations of unique squares adding up to 13. The only solution is $2^2 + 3^2$.

Function Description

Complete the powerSum function in the editor below. It should return an integer that represents the number of possible combinations.

powerSum has the following parameter(s):

- X: the integer to sum to
- N: the integer power to raise numbers to

Input Format

NEED HELP?

📖 View discussions

📖 View editorial

🏆 View top submissions

RATE THIS CHALLENGE

★ ★ ★ ★ ★

MORE DETAILS

represents the number of possible combinations.

powerSum has the following parameter(s):

- X: the integer to sum to

- N: the integer power to raise numbers to

Input Format

The first line contains an integer X.

The second line contains an integer N.

Constraints

- $1 \leq X \leq 1000$
- $2 \leq N \leq 10$

Output Format

Output a single integer, the number of possible combinations caclulated.

Sample Input 0

```
10
2
```

Output a single integer, the number of possible combinations caclulated.

Sample Input 0

```
10
2
```

Sample Output 0

```
1
```

Explanation 0

If $X = 10$ and $N = 2$, we need to find the number of ways that 10 can be represented as the sum of squares of unique numbers.

$$10 = 1^2 + 3^2$$

This is the only way in which 10 can be expressed as the sum of unique squares.

Sample Input 1

Sample Input 1

```
100
2
```

Sample Output 1

```
3
```

Explanation 1

$$100 = (10^2) = (6^2 + 8^2) = (1^2 + 3^2 + 4^2 + 5^2 + 7^2)$$

Sample Input 2

```
100
3
```

Sample Output 2

- $1 \leq X \leq 1000$
- $2 \leq N \leq 10$

Output Format

Output a single integer, the number of possible combinations caclulated.

Sample Input 0

```
10
2
```

Sample Output 0

```
1
```

Explanation 0

If $X = 10$ and $N = 2$, we need to find the number of ways that 10 can be represented as the sum of squares of unique numbers.

$10 = 1^2 + 3^2$

Explanation 1

$$100 = (10^2) = (6^2 + 8^2) = (1^2 + 3^2 + 4^2 + 5^2 + 7^2)$$

Sample Input 2

```
100
3
```

Sample Output 2

```
1
```

Explanation 2

100 can be expressed as the sum of the cubes of $1, 2, 3, 4$.

$(1 + 8 + 27 + 64 = 100)$. There is no other way to express 100 as the sum of cubes.

hacker rank power sum

how to solve this
problem ?

i need to try all possible
combinations

let's think on paper on what i need to create

power(1,n) or 0 + power(2,n) or 0 + power(3,n) or 0 + power (4,n) or 0
etc .. ....

what's our recursion paramters ?

i represents our 1 and 2 and 3 and 4 etc ..
and sum represents the summation

hacker rank power sum

input : n and x   constrains : x from 1 to 1000 , n from 2 to 10

power(1,n) or 0 + power(2,n) or 0 + power(3,n) or 0 + power (4,n) or 0
etc .. ....

what's our recursion paramters ?

recu(int i,int sum)

recu(i+1,sum+power(i,n))
recu(i+1,sum)

i represents our 1 and 2 and 3 and 4 etc ..
and sum represents the summation

if (i==50)
{ if (sum==x){cnt++} return;}

hacker rank power sum

```
ll n,x;
int cnt;
void solve(ll i,ll sum)
{
    if (i==50){if (sum==x){cnt++;}return ;}
    if (sum+fastpower(i,n)<=x){solve(i+1,sum+fastpower(i,n));}
    solve(i+1,sum);

}
```

```
cin>>x>>n;
solve(1,0);
cout<<cnt;
return 0;
```

solve(1,0)

x=10

n=2

solve(2,1)   **solve(2,0)**

solve(3,5)   **solve(3,1)**   **solve(3,4)**   **solve(3,0)**

solve(4,5) **solve(4,10)** **solve(4,1)** **solve(4,4)** **solve(4,9)** **solve(4,0)**