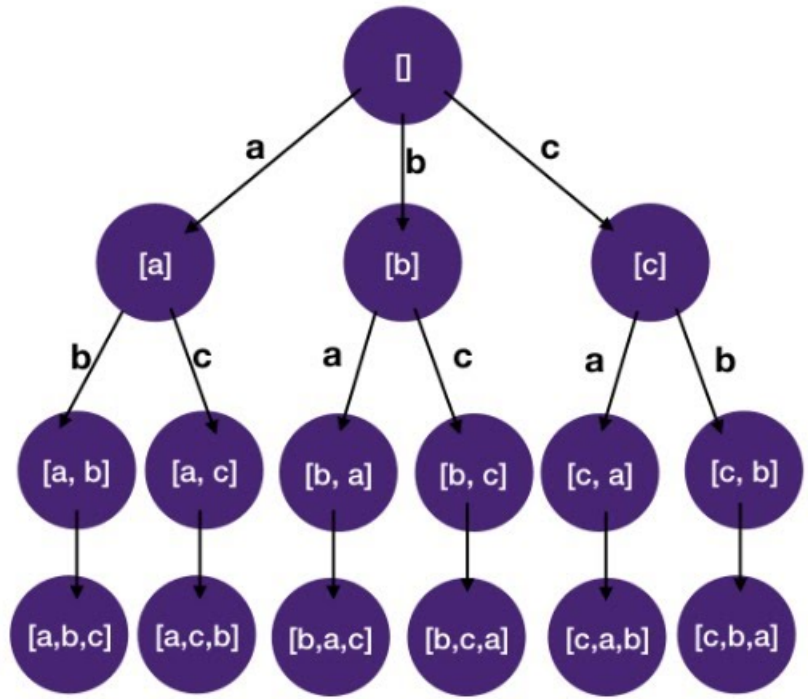
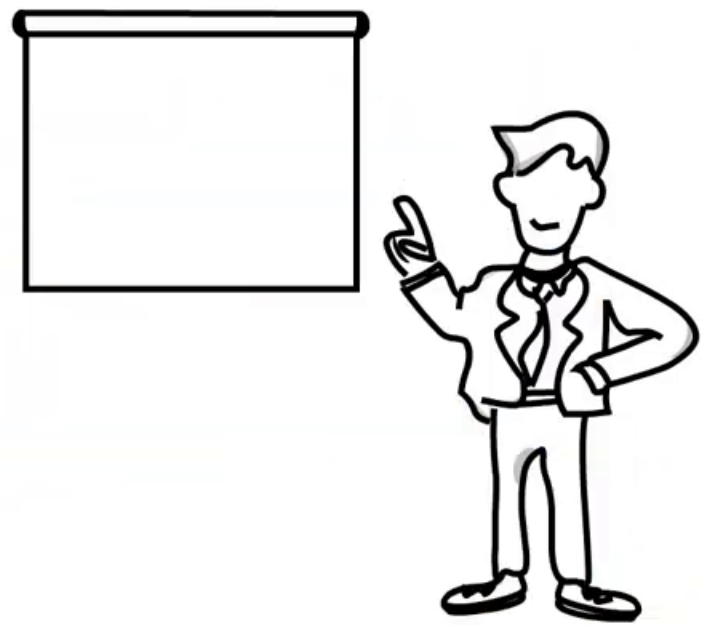
Recursive Examples

(binary strings) (print all strings of length k)



prior knowledge

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



Recursive Examples

Given a number n, find all binary sequences of length 2n such that sum of first n bits is same as sum of last n bits.

Input: N = 2

Output:

0101

1111

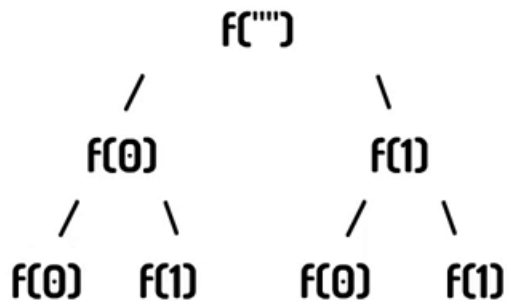
1001

0110

0000

1010

```
int n;
void recu(string s,int sum,int sum2)
{
    if (s.size()==2*n)
    {
        if (sum==sum2){cout<<s<<endl;}
        return ;
    }
    recu(s+'0',sum,sum2);
    if (s.size()<n){recu(s+'1',sum+1,sum2);}
    else {recu(s+'1',sum,sum2+1);}
}
```



till the base case stops the tree

Recursive Examples

Print all possible strings of length k that can be formed from a set of n characters

Input:

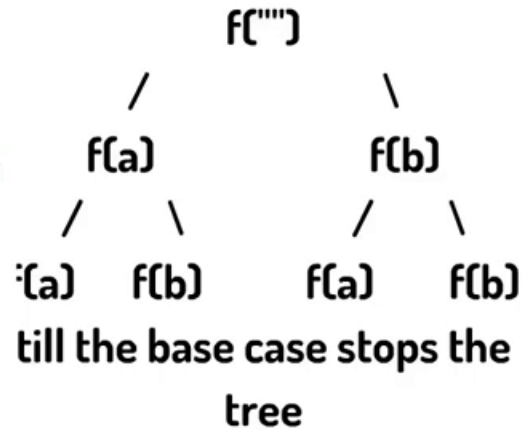
set[] = {'a', 'b'}, k = 3

Output:

aaa
aab
aba
abb
baa
bab
bba
bbb

```
int k;
void recu2(string s)
{
    if (s.size()==k){cout<<s<<endl;return ;}

    for (int i=0;i<2;i++)
    {
      recu2(s+arr[i]);
    }
 }
```



Recursion Tree Visualizer

recursion.vercel.app

Pre-defined templates

Binomial Coefficient

Global variables

=

=

Recursive function

python

```
def fn(n,k):
  # given n items, how many different
possible subsets of k items can be formed

  if (k == 0 or n == k):
    return 1

  return fn(n-1, k-1) + fn(n-1, k)
```

Options

Enable step-by-step animation

Enable memoization

Enable dark mode

fn(5,2)

Run

fn(5,2) starts running

5,2

Type here to search

92°F Mostly cloudy

ENG

2:27 PM
7/25/2021

Recursion Tree Visualizer

recursion.vercel.app

Pre-defined templates

Binomial Coefficient

Global variables

=

=

Recursive function

python

```
def fn(n,k):
  # given n items, how many different
possible subsets of k items can be formed

  if (k == 0 or n == k):
    return 1

  return fn(n-1, k-1) + fn(n-1, k)
```

Options

Enable step-by-step animation

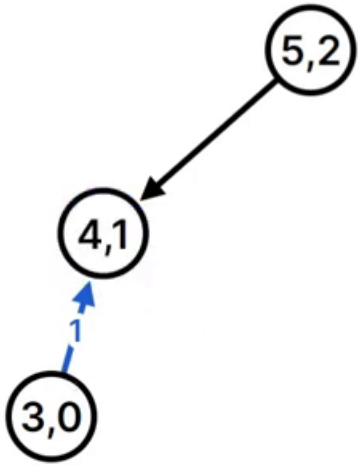
Enable memoization

Enable dark mode

fn(5,2)

Run

fn(3,0) returns 1 to fn(4,1)



Type here to search

92°F Mostly cloudy

ENG

2:27 PM
7/25/2021

Recursion Tree Visualizer

recursion.vercel.app

Pre-defined templates

Binomial Coefficient

Global variables

=

=

Recursive function

python

```
def fn(n,k):
  # given n items, how many different
possible subsets of k items can be formed

  if (k == 0 or n == k):
    return 1

  return fn(n-1, k-1) + fn(n-1, k)
```

Options

Enable step-by-step animation

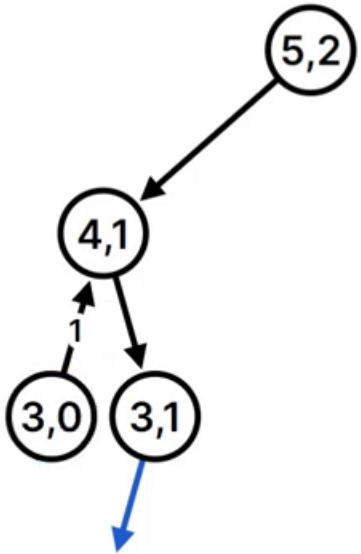
Enable memoization

Enable dark mode

fn(5,2)

Run

fn(3,1) calls fn(2,0)



Type here to search

92°F Mostly cloudy

ENG

2:28 PM
7/25/2021

Recursion Tree Visualizer

recursion.vercel.app

Pre-defined templates

Binomial Coefficient

Global variables

=

=

Recursive function

python

```
def fn(n,k):
  # given n items, how many different
possible subsets of k items can be formed

  if (k == 0 or n == k):
    return 1

  return fn(n-1, k-1) + fn(n-1, k)
```

Options

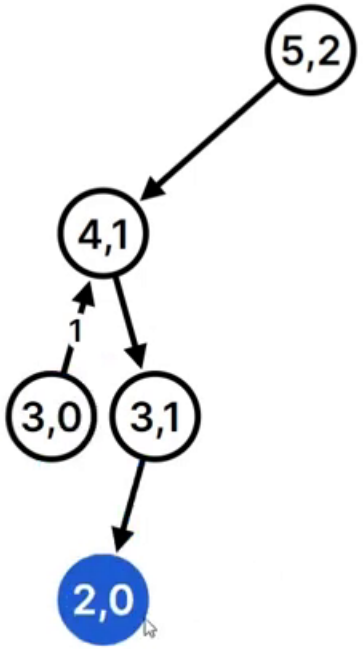
Enable step-by-step animation

Enable memoization

Enable dark mode

fn(5,2) Run

fn(2,0) starts running



Type here to search

92°F Mostly cloudy ENG 2:28 PM 7/25/2021

Recursion Tree Visualizer

recursion.vercel.app

Pre-defined templates

Binomial Coefficient

Global variables

=

=

Recursive function

python

```
def fn(n,k):
  # given n items, how many different
possible subsets of k items can be formed

  if (k == 0 or n == k):
    return 1

  return fn(n-1, k-1) + fn(n-1, k)
```

Options

Enable step-by-step animation

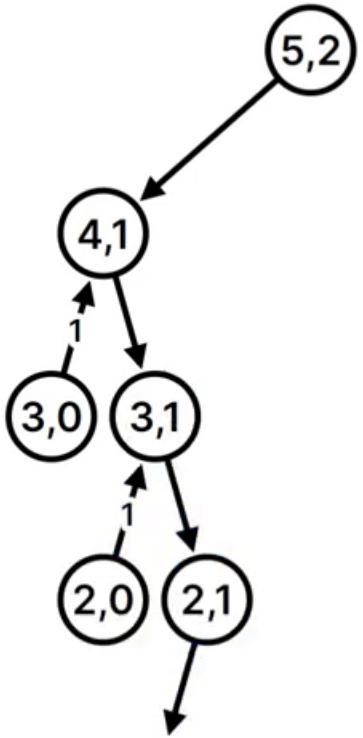
Enable memoization

Enable dark mode

fn(5,2)

Run

fn(1,0) starts running



Type here to search

92°F Mostly cloudy

ENG

2:28 PM
7/25/2021

Recursion Tree Visualizer

recursion.vercel.app

Pre-defined templates

Binomial Coefficient

Global variables

=

=

Recursive function

python

```
def fn(n,k):
  # given n items, how many different
possible subsets of k items can be formed

  if (k == 0 or n == k):
    return 1

  return fn(n-1, k-1) + fn(n-1, k)
```

Options

Enable step-by-step animation

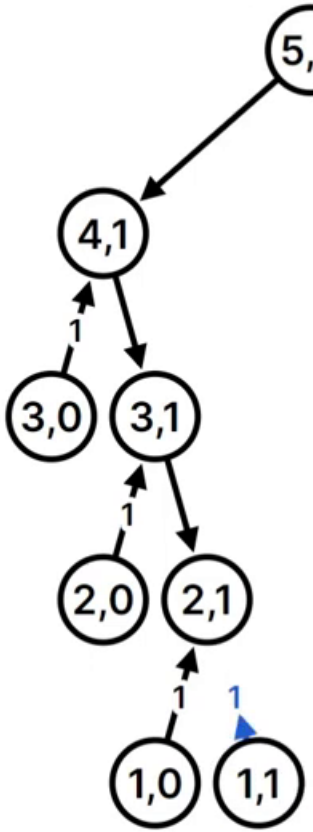
Enable memoization

Enable dark mode

fn(5,2)

Run

fn(1,1) returns 1 to fn(2,1)



Type here to search

92°F Mostly cloudy

ENG

2:28 PM
7/25/2021

Recursion Tree Visualizer

recursion.vercel.app

Pre-defined templates

Binomial Coefficient

Global variables

=

=

Recursive function

python

```
def fn(n,k):
  # given n items, how many different
possible subsets of k items can be formed

  if (k == 0 or n == k):
    return 1

  return fn(n-1, k-1) + fn(n-1, k)
```

Options

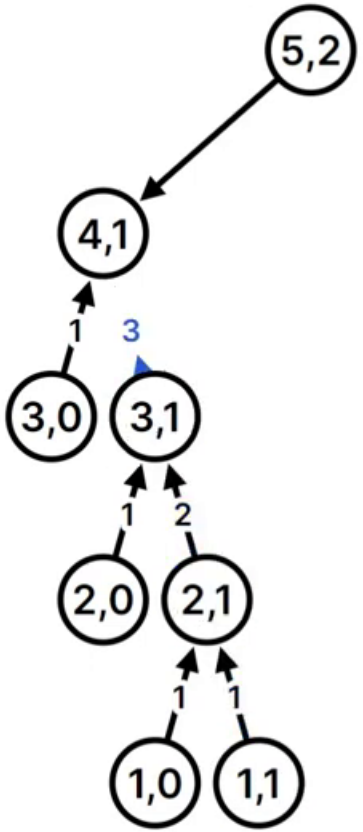
Enable step-by-step animation

Enable memoization

Enable dark mode

fn(5,2) Run

fn(3,1) returns 3 to fn(4,1)



Type here to search

92°F Mostly cloudy ENG 2:28 PM 7/25/2021

Recursion Tree Visualizer

recursion.vercel.app

Pre-defined templates

Binomial Coefficient

Global variables

=

=

Recursive function

python

```
def fn(n,k):
  # given n items, how many different
possible subsets of k items can be formed

  if (k == 0 or n == k):
    return 1

  return fn(n-1, k-1) + fn(n-1, k)
```

Options

Enable step-by-step animation

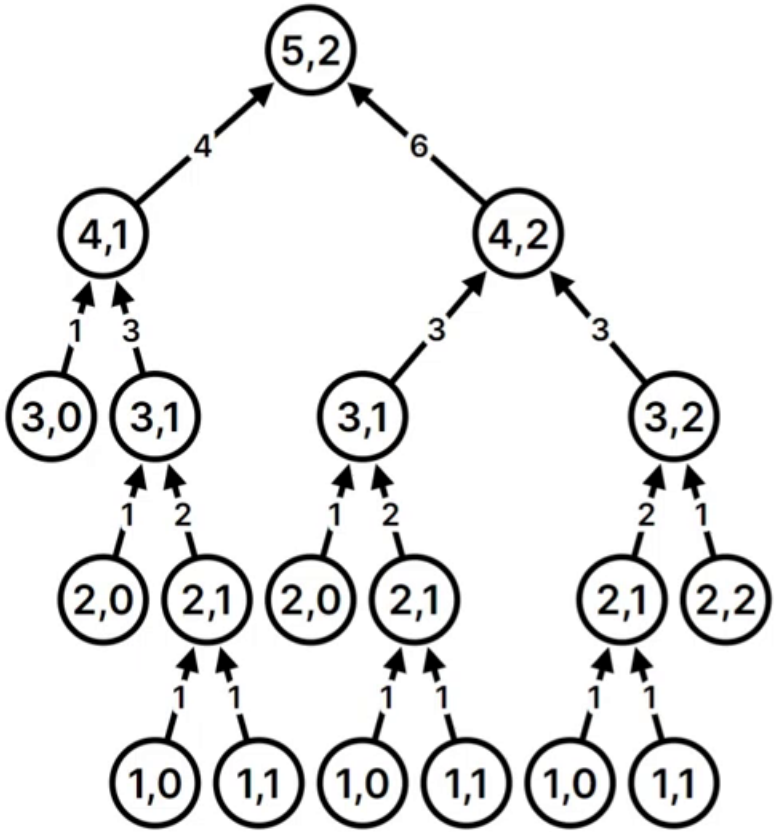
Enable memoization

Enable dark mode

fn(5,2)

Run

fn(5,2) returns 10



Type here to search

92°F Mostly cloudy

ENG

2:26 PM
7/25/2021