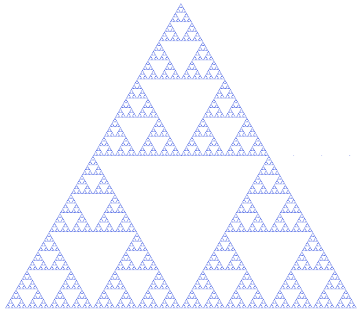
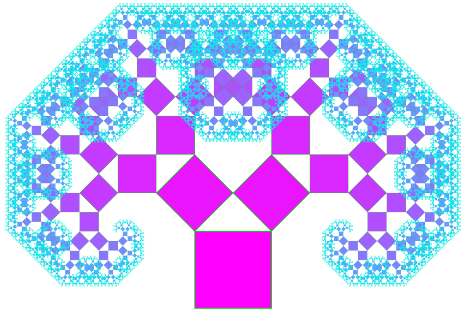
Recursion

Jiaxiang Zhang





Outline

- Key concepts – a toy example
- More advanced cases
- Recursion vs. iteration
- Pros and Cons of recursion

What is recursion?

- Stacking dolls with 6 sizes -- how to get the smallest one?

- Solve a big problem by solving **smaller or easier** problems of the **same** type

How to implement recursion?

1. Recursive calls

In [ ]:

```
def getDoll():
  print('Open a doll')
  return getDoll()        # Recursive call

getDoll()
```

How to implement recursion?

1. Recursive calls
    - Addressing a smaller/simpler problem in each recurvive call

In [ ]:

```
def getDoll(n):
  print('Open a doll, size =',n)
  return getDoll(n-1)       # Recursive call to solve the smaller problem

getDoll(6)
```

How to implement recursion?

1. Base case (stopping condition)
    - The smallest/easiest problem
    - The recursive call ends here

In [ ]:

```
n=1

if (n==1):
  print('Found the smallest doll, size =',n)
```

Recursion = recursive calls + base case(s)

In [ ]:

```
def getDoll(n):
  if (n==1):    # Base case
    print('Found the smallest doll, size =',n)
  else:
    print('Open a doll, size =',n)
    return getDoll(n-1)   # Recursive call to solve the smaller problem

getDoll(6)
```

Outline

- Key concepts – a toy example
- More advanced cases
    1. Fractional
    2. Binary search
    3. Tower of Hanoi
- Recursion vs. iteration
- Pros and cons of recursion

Factorial of N

Normal solution

$$N! = f(N) = \begin{cases} 1 & \text{if } N = 0 \\ N * \underline{(N-1) * (N-2) * \cdots * 2 * 1} & \text{if } N > 0 \end{cases} \tag{1}$$

Recursive solution

$$N! = f(N) = \begin{cases} 1 & \text{if } N = 0, \text{(Base case)} \\ N * \underline{f(N-1)} & \text{if } N > 0, \text{(Recursive call)} \end{cases} \tag{2}$$

Factorial of N - recursive solution

In [ ]:

```
def f(N):
  print('calculating the fractorial of',N)
  # Base case
  if (N==0):
    print('Reach the base case')
    return 1
  # Recursive call to solve the smaller problem
  else:
    f_n_1=N * f(N-1)
    print('Obtained the fractorial of',N-1)
    return f_n_1

N=4; print('The factorial of',N,'is', f(N))
```