

написать рекурсивное вычисление чисел фибоначчи

1 1 2 3 5 8 13 21

1 2 3 4 5 6 7

fib(1)=1

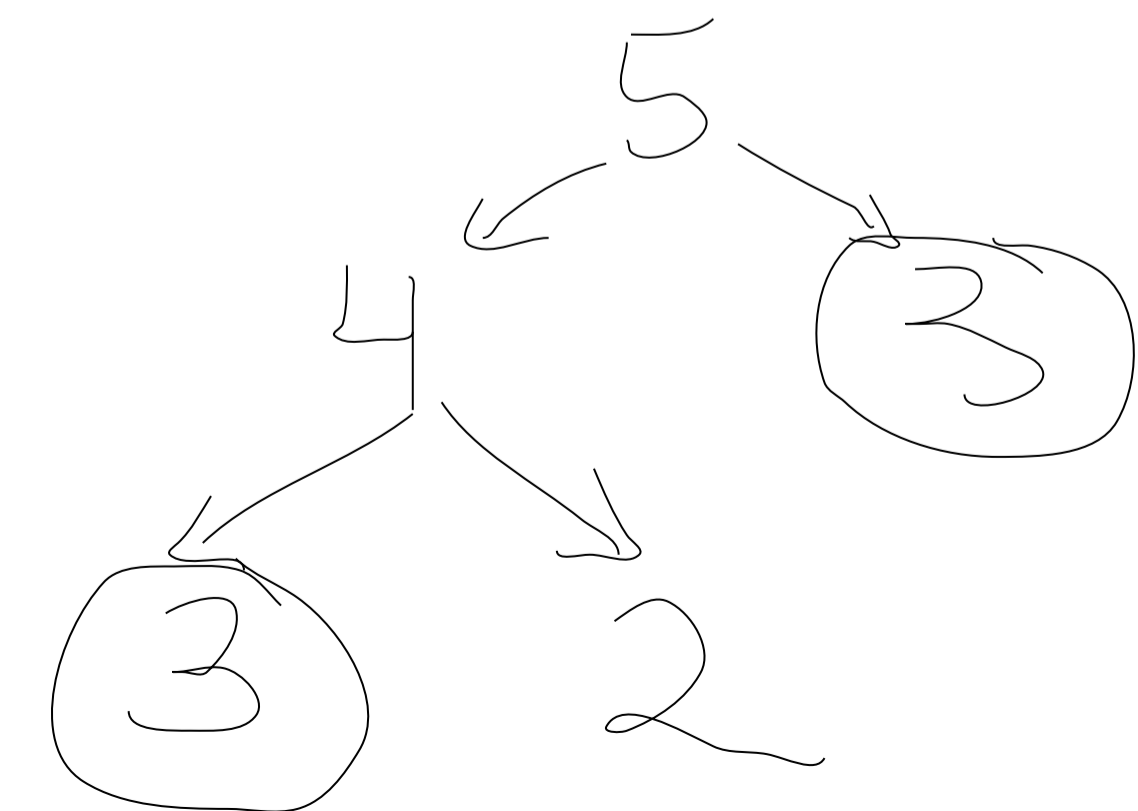
fib(2)=1

fib(3)=2

....

fib(n)=fib(n-1)+fib(n-2)

inf fib(int n)//7



```
def fib(k):  
    if k==1 or k==2:  
        return 1  
    elif k==3:  
        return 2  
    else:  
        x=fib(k-1)+fib(k-2)  
    return x  
k=int(input())  
print(fib(k))
```

```
k=int(input())  
ms=[1,1,2]  
i=3  
while i<=k-1:  
    ms.append(ms[i-1]+ms[i-2])  
    i+=1  
print(ms[k-1])
```

```
def fib(k,ms):  
    if k not in ms:  
        if k==1 or k==2:  
            ms[k]=1  
            return 1  
        elif k==3:  
            ms[k]=2  
            return 2  
        else:  
            x=fib(k-1,ms)+fib(k-2,ms)  
            ms[k]=x  
            return x  
    else:  
        return ms[k]  
k=int(input())  
ms=dict()  
print(fib(k,ms))
```

рекурсивное соотношение

каждый промежуточный
результат вычисляется
ровно 1 раз

Считается основную
задачу решать не умеем
 $F(n)$, но мы ее разбиваем
на маленькие подзадачи,
которые решать умеет с
помощью рекурсивного
соотношения