Recursion

Αναδρομή

Christos Katsoulas

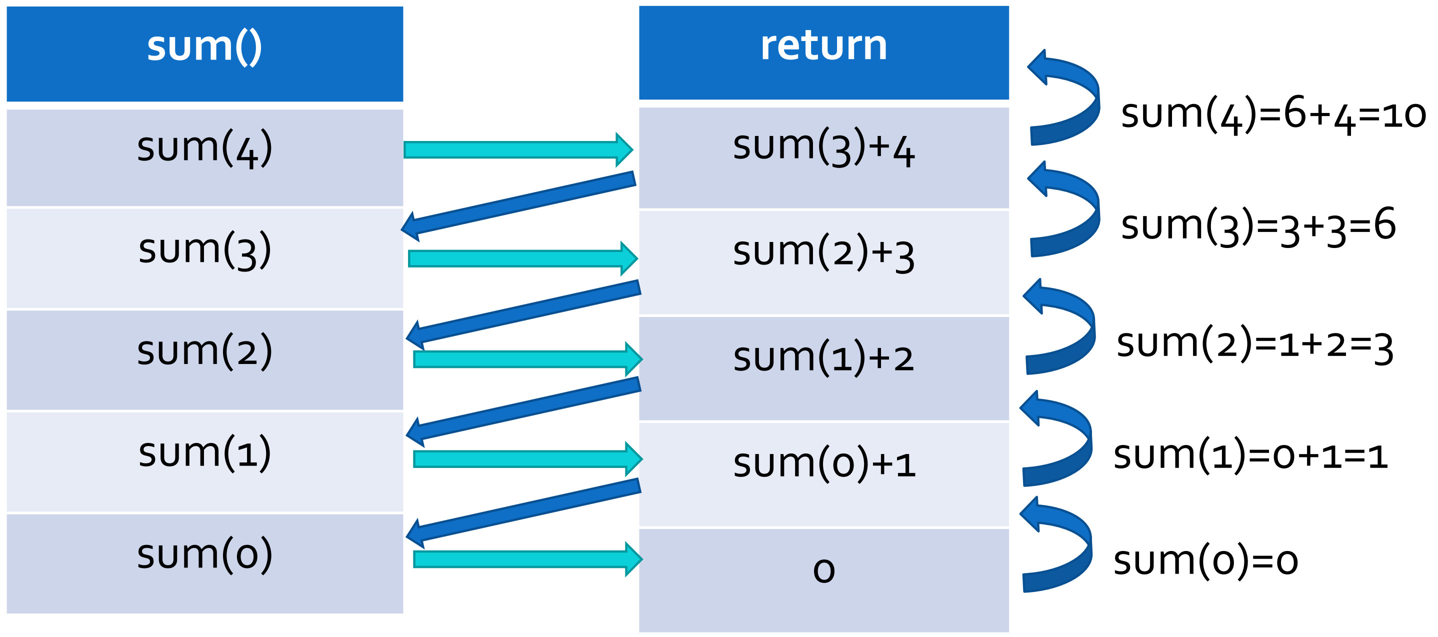
What is Recursion?

- A function calls itself.
- Central idea of Computer Science.
- The solution to a problem depends on solutions to smaller instances of the same problem.
- Recursion can solve the problems without iterative control structures (“while” or “for”).

```
void countDown(int counter){
        if counter==0 {
                return;
        }
        else {
                cout<<counter<<endl;
                countDown(counter-1);
         }
}
int main(){
        int num=8;
        countdown(num);
}
```

Sum from zero

```
int sumFromZero (int n){
    if (n == 0) return 0;
    else return sum(n - 1) + n;
}
```



sum()	return
sum(4)	sum(3)+4
sum(3)	sum(2)+3
sum(2)	sum(1)+2
sum(1)	sum(0)+1
sum(0)	0

sum(4)=6+4=10

sum(3)=3+3=6

sum(2)=1+2=3

sum(1)=0+1=1

sum(0)=0

Fibonacci numbers

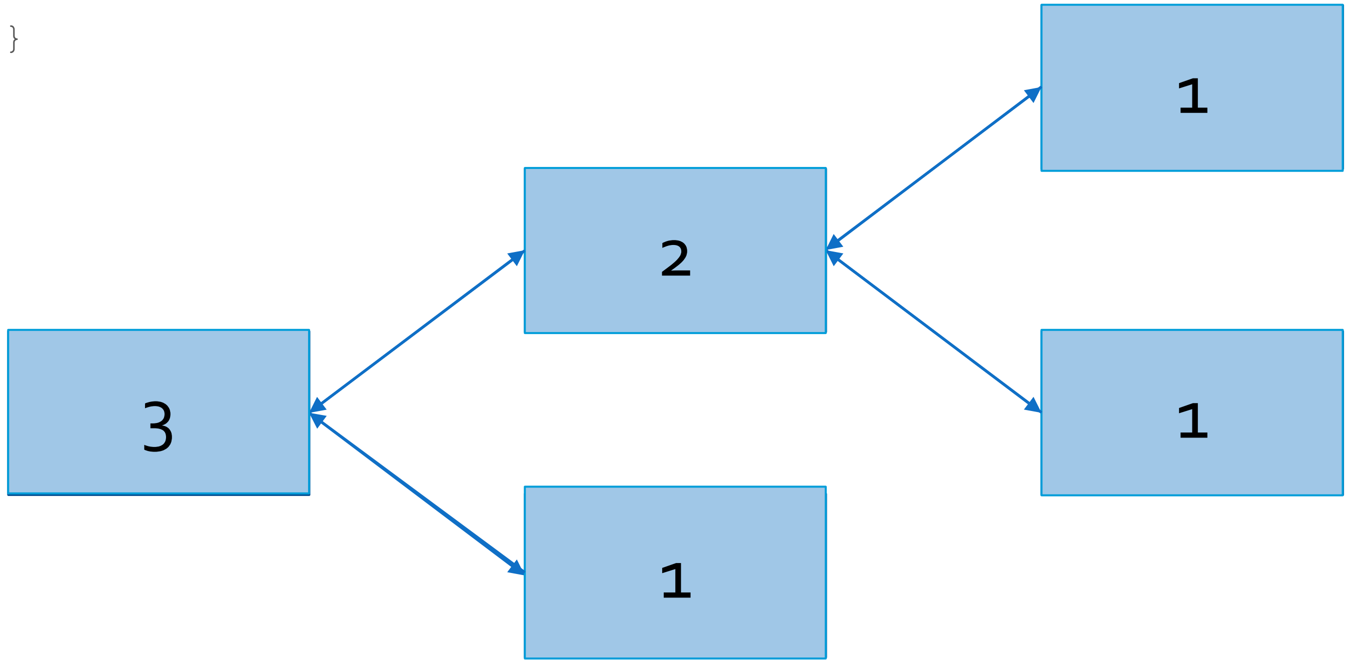
- The first two numbers are 0 and 1.
- Each subsequent number is the sum of the previous two.
- The sequence F_n of Fibonacci numbers is defined by the recurrence relation:
- $F_n = F_{n-1} + F_{n-2}$
- $F_0 = 0$
- $F_1 = 1$

```
int Fib(n){
        if(n == 1 || n == 2){
            return 1;
        }
        else{
            return Fib(n-1)+Fib(n-2)
        }
}
```

How the Fibonacci numbers recursion works

```
int Fib(n){
   if(n == 1 || n == 2) return 1;
   else return Fib(n-1)+Fib(n-2);
}
```

Calculate Fibonacci (4)



Fib (4) = 3

Simple recursive functions

- Count Down
- Sum from zero (or any other number)
- Factorial
- Greatest Common Divider
- Towers of Hanoi

Towers of Hanoi

OBJECTIVE

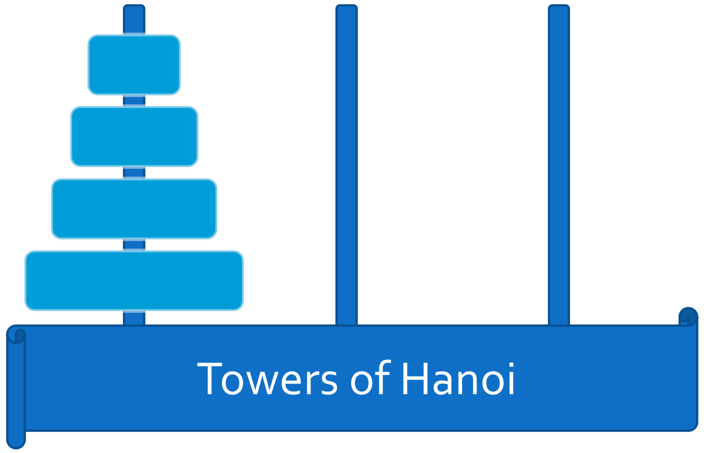
- Move the stack to the third rod.

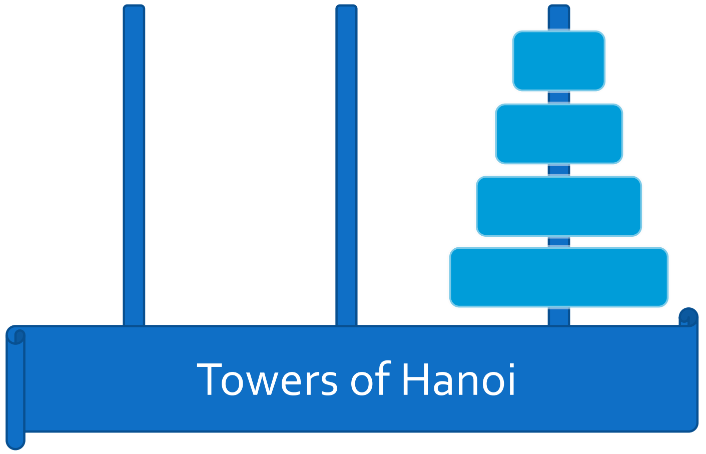
RULES

- Move one disk at a time.
- A disk can be moved only if it is the uppermost disk on a stack.
- No disk may be placed on top of a smaller disk.

RECURSION

- The problem can be solved by breaking it down to smaller problems until a solution is reached.





Recursion requires Abstraction

IDEA

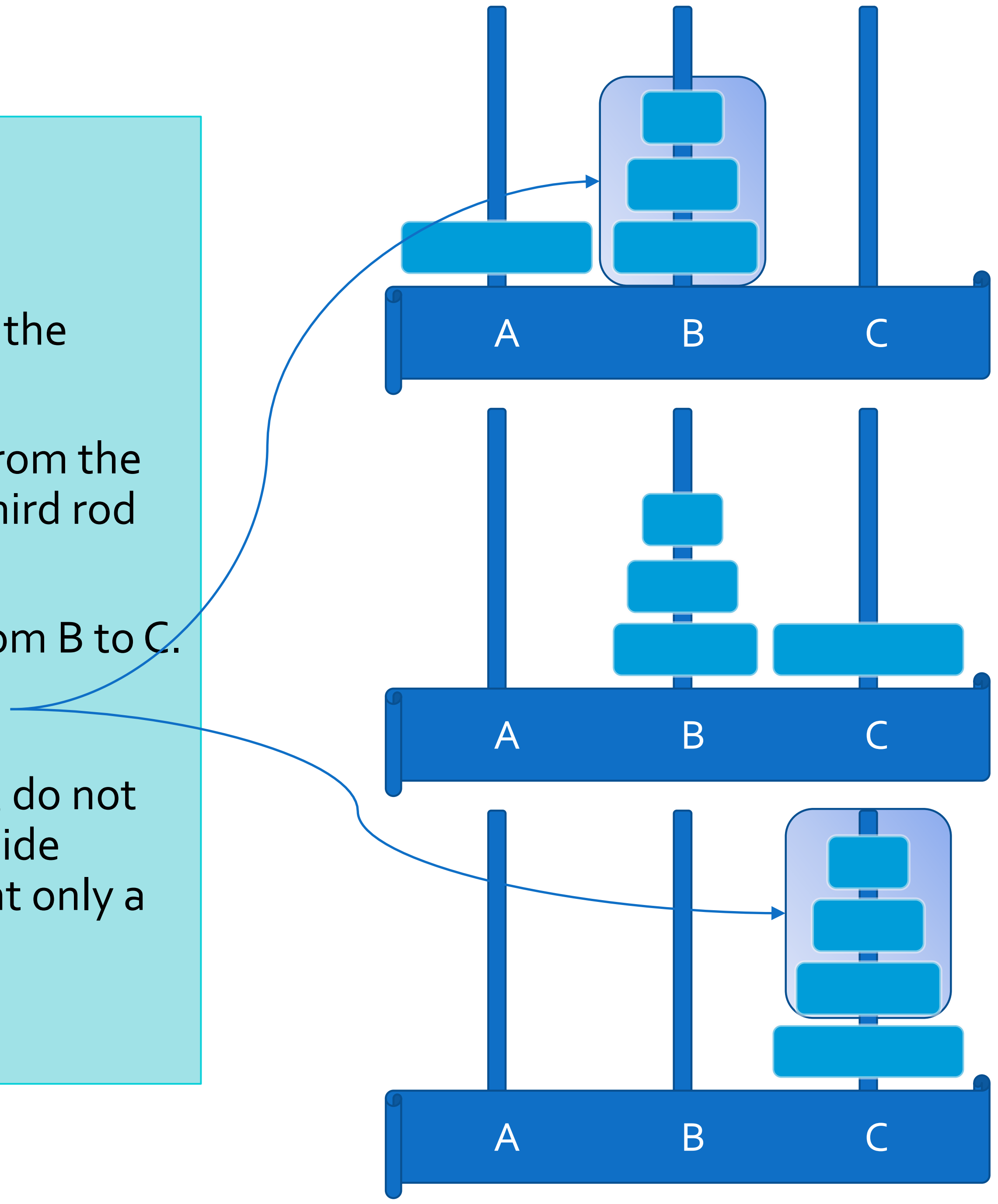
1. Move n-1 disks to the second rod (B).

2. Move the n disk from the first rod (A) to the third rod (C).

3. Move n-1 disks from B to C.

ABSTRACTION

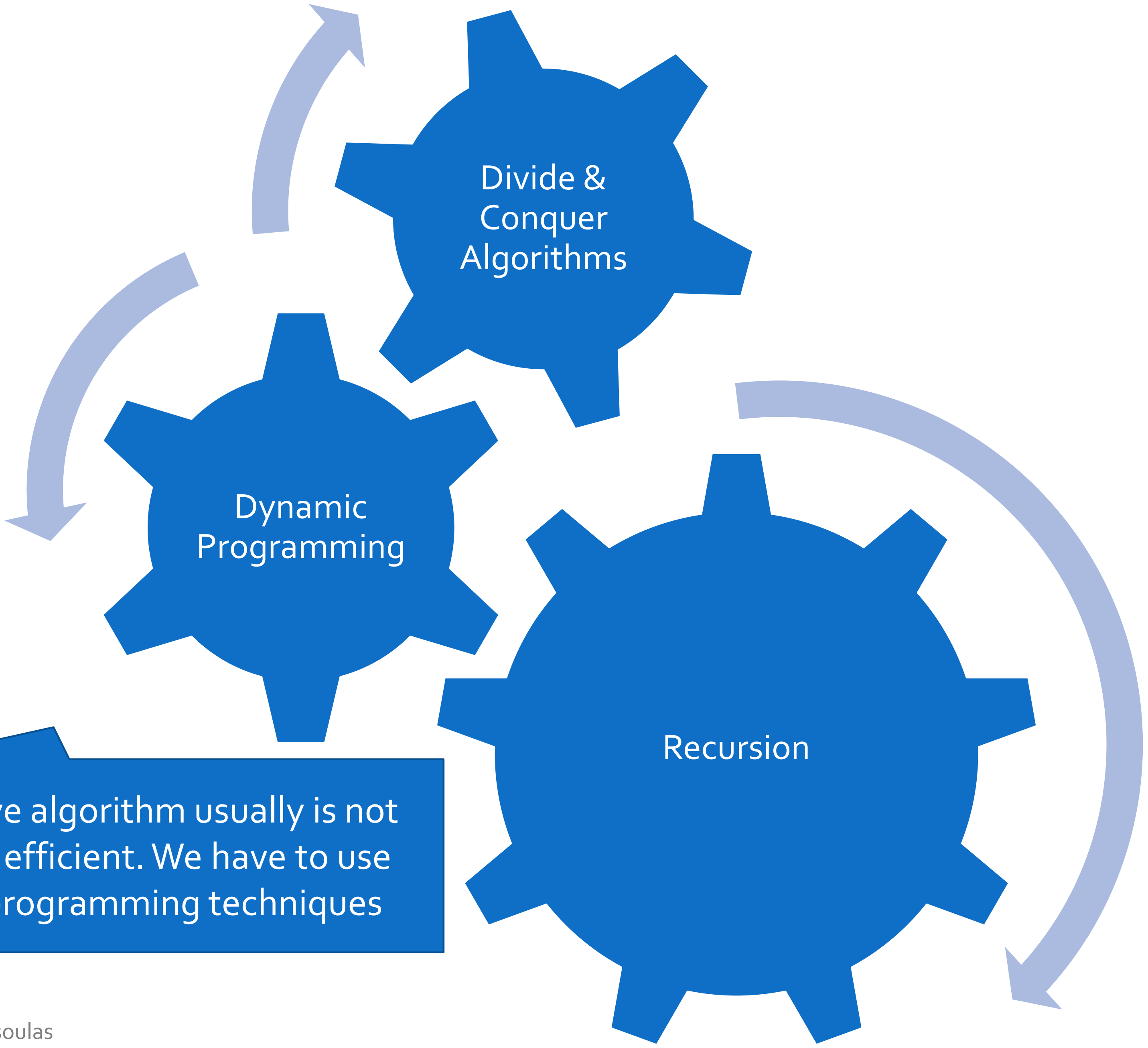
During steps 1 and 3 do not mess with details. Hide details and represent only a certain intention.



Recursion

Divide & Conquer Algorithms

Dynamic Programming



A recursive algorithm usually is not the most efficient. We have to use further programming techniques