遞迴函式

Recursive function

Ya-Chun Hsu @ Taipei Municipal Jianguo High School

遞迴 recursion

- 知道如何解比較簡單的狀況
- 較複雜的狀況可以用較簡單的狀況推出來
- 遞迴函式會自己呼叫自己（直接或間接）

n階層

$$n! = \begin{cases} 1 & , if\ n = 0 \\ n * (n-1)! & , if\ n > 0 \end{cases}$$

n階層

```
4  int factorial(int n){
5      int val=1;
6      for(int i=2;i<=n;i++){
7          val*=i;
8      }
9      return val;
10 }
```

使用迭代(非遞迴)方式定義函式

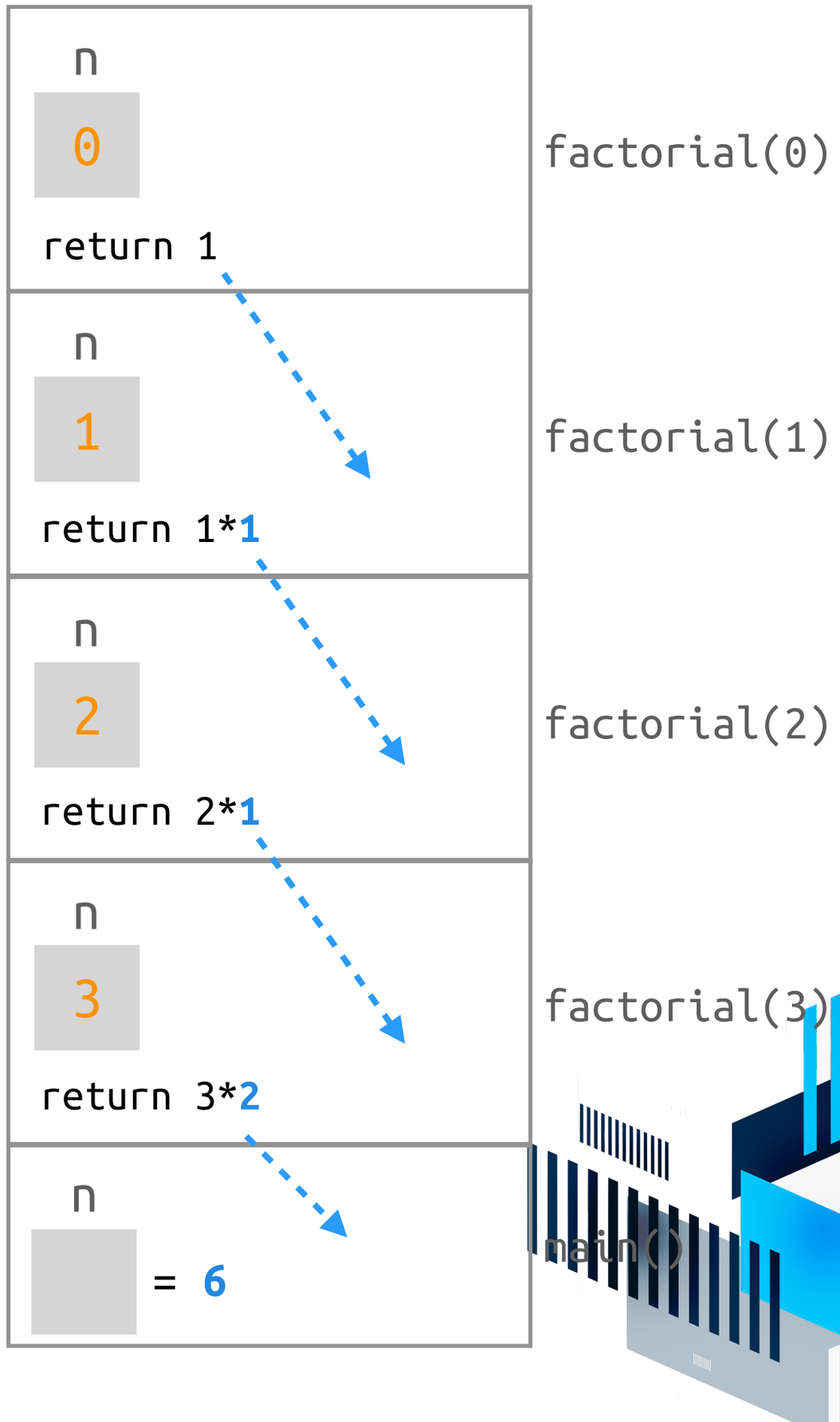
or

```
4  int factorial(int n){
5      if(n==0)
6          return 1;
7      else
8          return n*factorial(n-1);
9  }
```

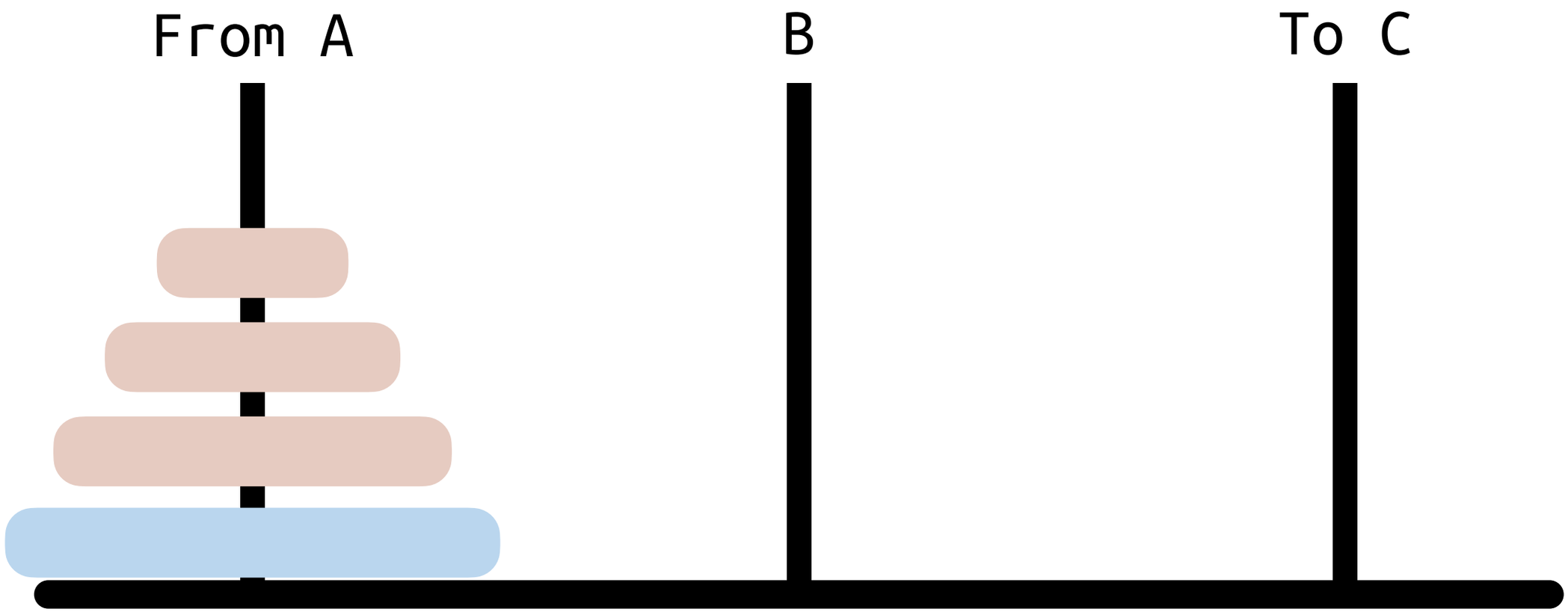
使用遞迴定義函式

n階層

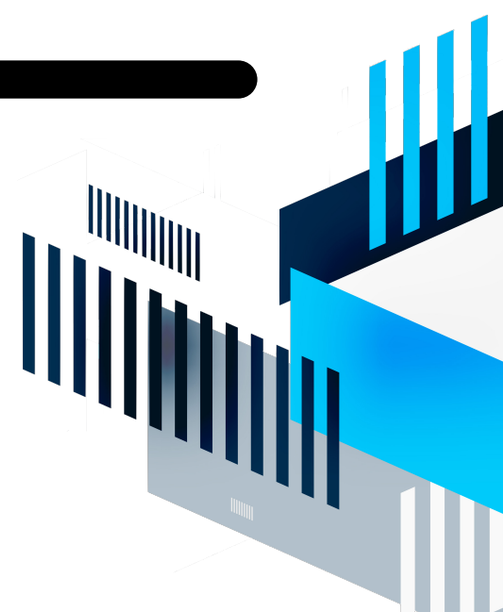
```cpp
#include <iostream>
using namespace std;

int factorial(int n){
    if(n==0)
        return 1;
    else
        return n*factorial(n-1);
}
int main(){
    int n=factorial(3);
    return 0;
}
```



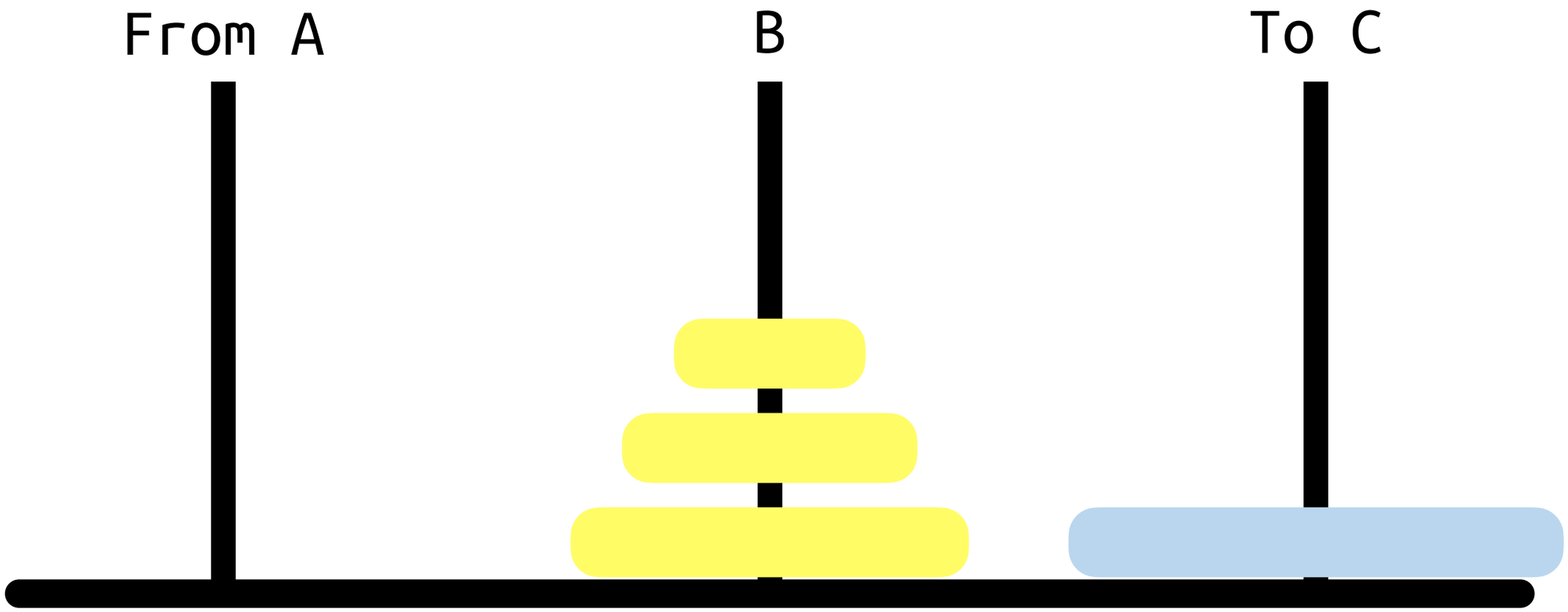
河內塔



從A移動4個盤子到C ＝ 從A移動3個盤子到B
從A移動1個盤子到C
從B移動3個盤子到C



河內塔



從A移動4個盤子到C ＝

從A移動3個盤子到B

從A移動1個盤子到C

從B移動3個盤子到C

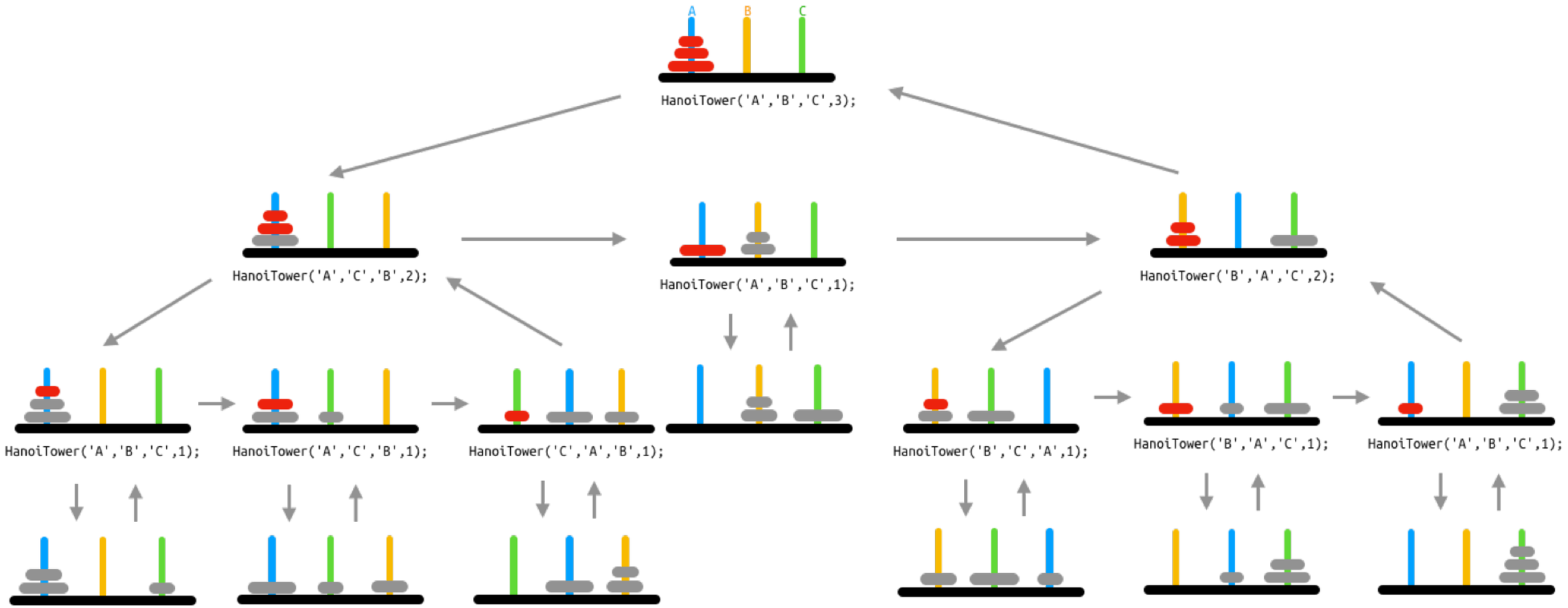
河內塔

```cpp
#include <iostream>
using namespace std;

void HanoiTower(char from,char buf,char to,int num_disk){
    if(num_disk==1)
        cout<<"Move a disk from "<<from<<" to "<<to<<endl;
    else{
        HanoiTower(from,to,buf,num_disk-1);
        HanoiTower(from,buf,to,1);
        HanoiTower(buf,from,to,num_disk-1);
    }
}
int main(){
    HanoiTower('A','B','C',3);
    return 0;
}
```

Output

```
Move a disk from A to C
Move a disk from A to B
Move a disk from C to B
Move a disk from A to C
Move a disk from B to A
Move a disk from B to C
Move a disk from A to C
```

河內塔




```cpp
void HanoiTower(char from,char buf,char to,int num_disk){
    if(num_disk==1)
        cout<<"Move a disk from "<<from<<" to "<<to<<endl;
    else{
        HanoiTower(from,to,buf,num_disk-1);
        HanoiTower(from,buf,to,1);
        HanoiTower(buf,from,to,num_disk-1);
    }
}
```

遞迴

- 遞迴程式相當方便
- 但是，遞迴呼叫相當花時間和空間
- 所以，還是盡量使用迴圈

```
 4  void f(int n){
 5      if (n <= 0) return;
 6      cout<<n<<endl;
 7      f(n-1);
 8  }
 9  int main(){
10      f(10000000);
11      cout<<"Great!"<<endl;
12  }
```

在程式跑到Great之前就會先Stack Overflow