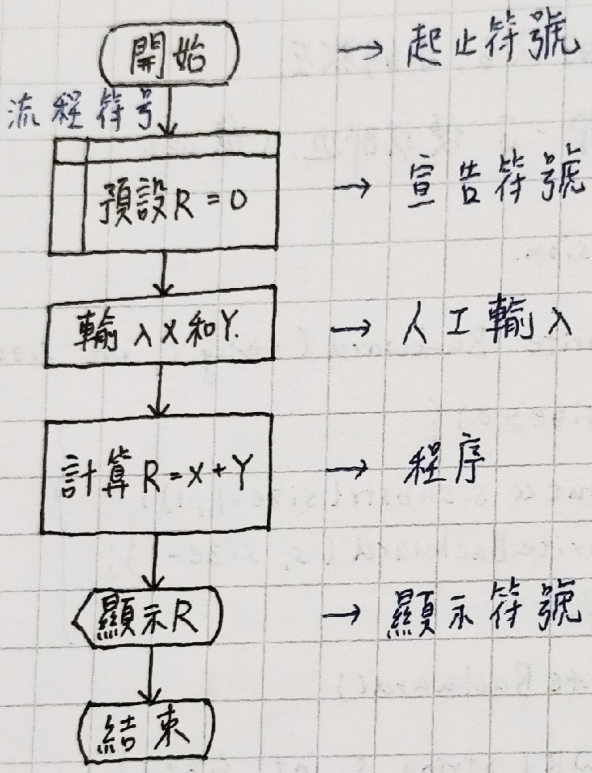
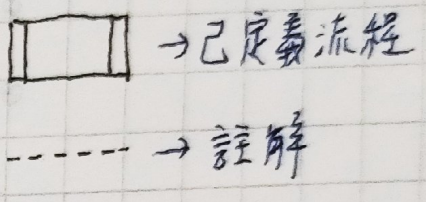
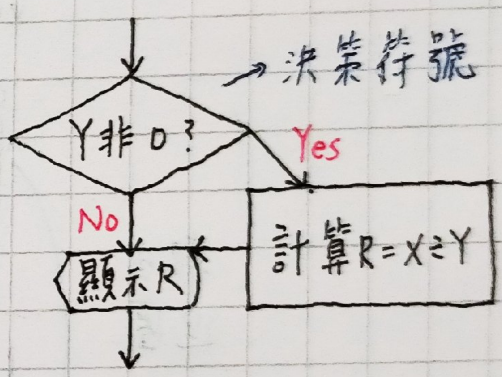
流程圖

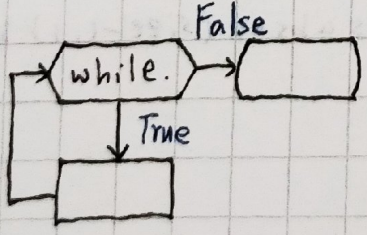
符號	字母代碼	描述
○	O	處理
IL	I	檢查
→	M	傳送
D	D	延遲
▽	S	儲存



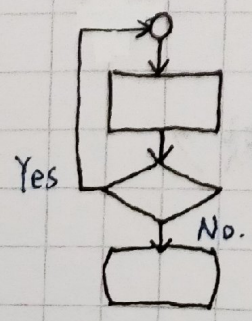




* while 迴圈.



* do-while 迴圈.

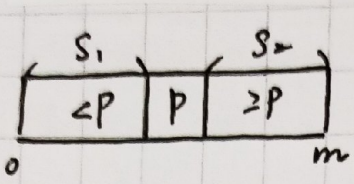


遞迴：

大 Q → 小 Q.

eg: 階乘. 最大公因數. 搜尋. 費式數列. 河內塔.

1. 找 k^{th} smallest item



- 把比P大的放右，小的放左.
- 看 k 在那一區. 便找那邊. (遞迴)

2. Tail Recursion.

```
void writeBackward(string s, int size){
    if(size>0){
        cout << s.substr(size-1,1);
        writeBackward(s, size-1);
    }// if
}// writeBackward()
```

轉成迴圈

```
void WB(string s, int size){
    while(size>0){
        cout << s.substr(size-1,1);
        --size;
    }// while
}// WB()
```

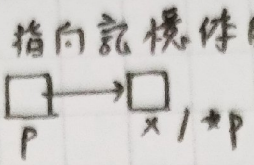
⇒效率↑

* 遞迴效率好壞⇒ 遞迴次數較少

指標 Pointer

→ int *p;

→ p = &x;



→ p = new int;

→ p = NULL;

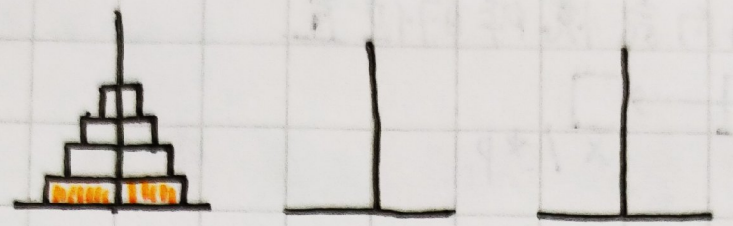
→ delete q; // 清空 memory.

q = NULL

动態陣列.

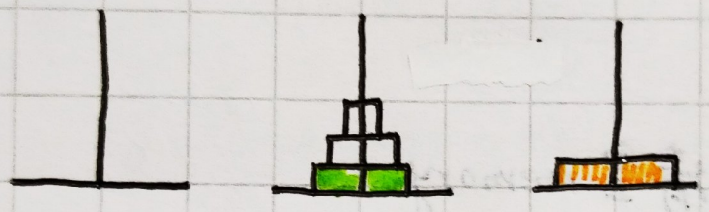
```
int ** array;
int * temp;.

    array = new int [sizey];.
    for ( int i=0; i<sizey; i++) {.
        temp = new int [sizex];
        array[i] = temp;
    } // for.
```

河内塔：n个盘子需2^n-1个步骤



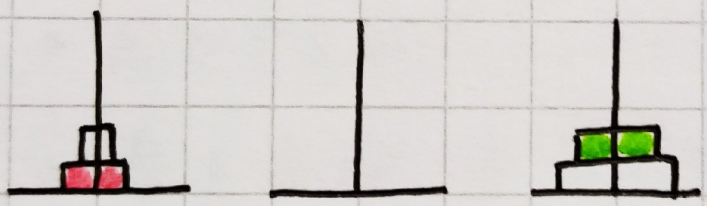
×15

=7×2+1



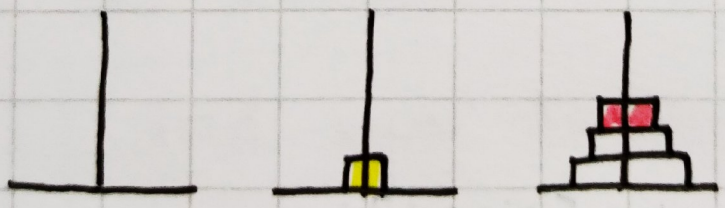
×2

=3×2+1



×3

=1×2+1



×1