迴圈 vs. 遞迴

iteration. recursion.

- Factorial 階乘 / Search in Arra
- Greatest Common Divisor 最大公因
- Fibonacci Series 費氏數列 /
- Tower of Hanoi 河內塔.

[Linear recursion 線性遞迴
Binary recursion. = 元遞迴

* writing a string backward.

base case : write the empty string backward.

條件：字串長度減 1.

① void writeBackward (string s, int size) {

```
if (size > 0) {
    cout << s.substr(size-1, 1); // output   (最後一個字)
    writeBackward(s, size-1); // recursive.
} // if
```

}.

輸出最後一個字元

Box trace. ex. ①

't' → s="cat" size=3 → 'a' → s="cat" size=2 → 't' → s="cat" size=1 → s="cat" size=0

② 輸出第一個字元

```
void writeBackward2(string s
        int size){
    if( s is empty).
    else {
        writeBackward2(...)
        cout << s.substr();
    } // else.
}.
```

c → s=cat → a → s=cat → ...

③

s=cat → s=at → s='t'.

✡ Greatest Common Divisor. (最大公因数)

- sol 1.

if (y==0) return x; | gcd1(x,y) = x. if y=0.
else if (y>x) return gcd1(x,y%x) | = gcd1(x, y mod x) if y>x.
else return gcd1(y,x%y) | = gcd1(y, x mody) otherwise.

base case: 其中一个整除的时候 (变0)

- sol 2.

if !(x%y) return y | gcd2(x,y) = y if x mod y=0.
else return gcd2(y,x%y) | = gcd(y, x mody) otherwise.

- sol 1 & sol 2. 用 box trace. 比较.

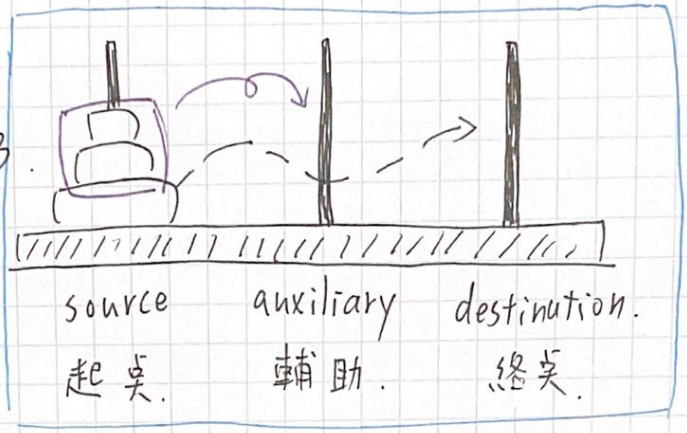
① [x=9 y=6 gcd1(6,3)=?] → [x=6, y=3 gcd1(3,0)=?] → [x=3, y=3 return 3;]

② [x=9 y=6 gcd2(6,3)=?] → [x=6 y=3 return 3;]

Towers of Hanoi. 河内塔.

n. 個数=3.

次数 2^n-1




```
SolveTowers ( count, source, destination, spare)
    if (count is 1)  Move a disk directly from source
                        to destination .
    else {
        SolveTowers( count-1, source, spare, destination)
        SolveTowers(  1, source. destination, spare)
        SolveTowers( count-1, spare, destination, source)
    }
```

Binary Search with an Array

```
int binary search (const int anArray[], int first. int last, int value
{
    int index
    if (first > last) index = -1;
    else {
        int mid = (first + last)/2;
        if (value == anArray [mid]) index = mid;
        else if (value < anArray [mid]) index = binary Search (anArray.
                                                first, mid-1, value)
        else index = binary Search (anArray, mid+1, last. value);
    }
    return index;
}
```

△左半刃.

△右半刃

Another Binary Recursion.

- Linear recursion.

```
int sum(int a, int b){  // assume a<b.
    if (a==b)
        return a;
    return a+sum(a+1, b);
}// sum()
```

loop = 3

加法 = 7.

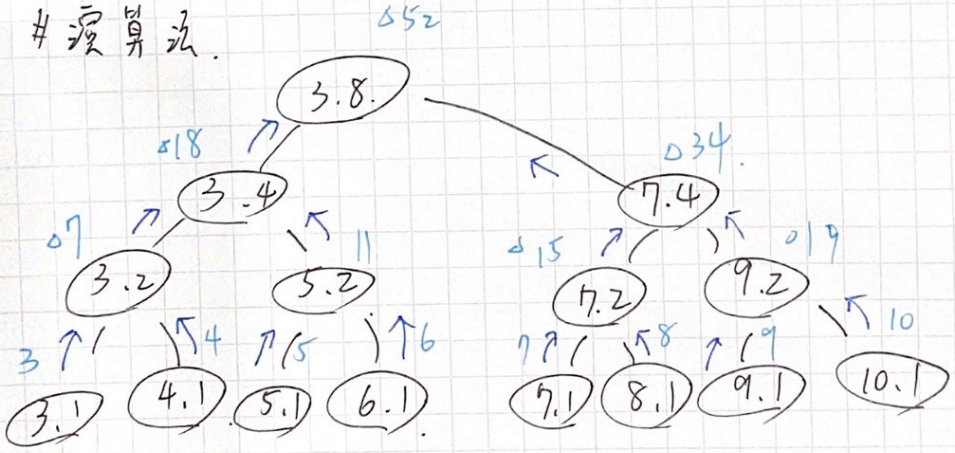
- Binary recursion.

```
int sum(int a, int n){  // assume n=b-a+1
    if (n==1)
        return a;
    return sum(a, n/2) + sum(a+n/2, n-n/2);
}// sum.
```

loop = 15

加法 = 7.

演算法.



Tail Recursion. (尾端遞迴)

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

Tail Recursion.

聰明的編譯器
自轉成迴圈 → 效率↑

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