NCKU Programming Contest Training Course
Course 1
2012/12/19

Sheng-Chi You (rabbit125)

jay_s6215@hotmail.com

Department of Computer Science and Information Engineering
National Cheng Kung University
Tainan, Taiwan

Outline

- C/C++ syntax introduction
- Steps to solve problems
- Time and Space Complexity Analysis
- Recursive

made by electron & kk & rabbit125

C/C++ syntax introduction

- Advance between C/C++ and JAVA
- Fast view for
  - Headers
  - variables
  - Iterations
  - Functions
  - Pointers
  - Coding style

made by electron & kk & rabbit125

C/C++ syntax introduction

- Headers
  - What are headers?
  - What kinds of headers do we need?
  - What is **using namespace std**?

```
#include <iostream>
#include <cstdio>
#include <cstdlib>
#include <cstring>
#include <cmath>
#include <algorithm>
using namespace std;
```

- We usually use following headers:
  - **iostream** - any I/O for C++ such as cin/cout
  - **cstdio** - printf() and scanf() in C/C++
  - **cstdlib** - some useful functions such as atoi(), malloc() or qsort() in C/C++
  - **cstring** - memset and string functions in C/C++
  - **cmath** - math functions in C/C++
  - **algorithm** - many useful functions such as max(), min() or sort() in C/C++

made by electron & kk & rabbit125

C/C++ syntax introduction

- Variables - Data types

Data types	byte	scanf	Range
int	4	%d	-2,147,483,648 to 2,147,483,647
unsigned int	4	%u	0 to 4294967295
long long int	8	%lld or %I64d	–9,223,372,036,854,775,808 to 9,223,372,036,854,775,807
float	4	%f	3.4E +/- 38 (7 digits)
double	8	%lf	1.7E +/- 308 (15 digits)
char	1	%c	-128 to 127
bool	1	N/A	true or false

made by electron & kk & rabbit125

C/C++ syntax introduction

Variables - Integers

- **int**
  - 4 bytes = 32 bits, range : -(2^31) ~ (2^31) – 1
- **unsigned int**
  - 4 bytes = 32 bits ,range : 0 ~ (2^32) – 1
  - Be careful ! Do not do this: for(unsigned i = 5; i >= 0; i--)
- **long long int**
  - 8 bytes = 64 bits ,range : -(2^63) ~ (2^63) – 1
  - Use %**I64d** in XP and both %**I64d** ,%**lld** in Vista and Linux, otherwise cin and cout
- **unsigned long long int**
  - 8 bytes = 64 bits ,range : 0~ (2^64) – 1
  - Use %**I64u** in XP and both %**I64u** ,%**llu** in Vista and Linux , otherwise cin and cout

```
int a = 46;
printf("%03d", a);  // 046
```


made by electron & kk & rabbit125

C/C++ syntax introduction

- Variables - floats
  - float
    - 4 bytes = 32 bits, range : -(2^31) ~ (2^31) – 1
    - 3.4E +/- 38 (7 digits)
  - double
    - 8 bytes = 64 bits ,range : 0 ~ (2^32) – 1
    - 1.7E +/- 308 (15 digits)
    - Use %.2lf can only print 2 digits after decimal, ex: `printf ( "%.2lf" , 5.126)// 5.13`

- Variables – bool
  - Bool
    - Only true and false
    - Cannot use scanf
    - Present 1 (true) in printf
    - Present 0 (false) in printf

```
bool a = false;
printf("%d", a);        // 0
```

```
bool a = 0;
printf("%d", a);        // 0
```

```
bool a = -1;
printf("%d", a);        // 1
```

```
bool a = 10;
printf("%d", a);        // 1
```


made by electron & kk & rabbit125

C/C++ syntax introduction

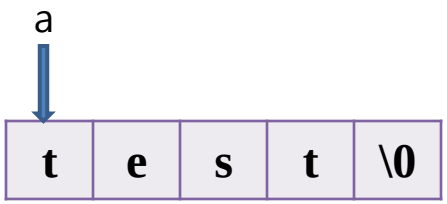
- Variables - char
  - char
    - 1 bytes = 8 bits, range : -(2^7) ~ (2^7) – 1
    - Using ASCII Code in C/C++
  - unsigned char
    - 1 bytes = 8 bits, range : 0 ~ (2^8) – 1
    - Seldom use
  - char[ ]
    - String type in C, can use %s
    - It will add a '/0' char at the last of string

```
char a = 'F' ;
printf("%c", a);          // F
```

```
char a = 'a' + 2;
printf("%c", a);          // c
```

```
char a = 65;
printf("%c", a);          // A
```

```
char a[5] = "test" ;
printf("%s", a);          // test
printf("%d", strlen(a));// 4
```



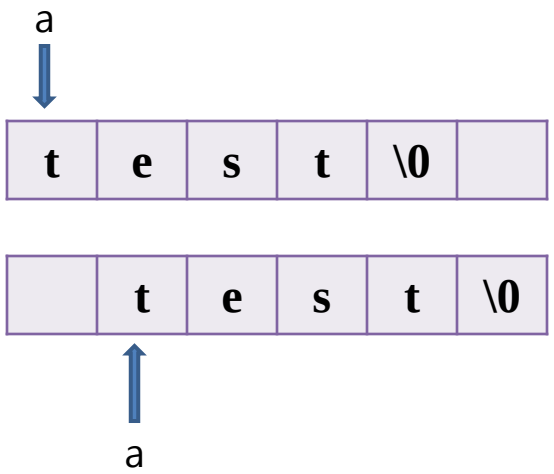
- PS: About ASCII, please refer to http://home.educities.edu.tw/wanker742126/asm/ap04.html

made by electron & kk & rabbit125

C/C++ syntax introduction

Variables – char cont.

- char[ ]
  - The declaration of char[ ]
  - Read string




```
char a[4] =  “test” ;      // illegal
```

```
char a[] =  “test” ;       // legal
```

```
char a[5];
a =  “test” ;              // illegal
```

```
char a[6];
scanf("%s", a);            // type  “test”

scanf("%s", a + 1);        // type  “test”
```


made by electron & kk & rabbit125

C/C++ syntax introduction

- Variables - global and local

```
int N = 100;

int main()
{
  printf("%d", N);     //100
  int N = 10;
  printf("%d", N);     //10
  return 0;
}
```

Scope of i

Scope of j

```
void function()
{
  for( int i = 1; i <= 5; i++)
  {
    for( int j = 1; j <= 10; j++)
    {
      printf("%d", i + j);
    }
    printf("%d", j);   // illegal ! ! !
  }
}
```


made by electron & kk & rabbit125

C/C++ syntax introduction

- Variables - structure
  - Using typedef or not

```
struct CAR
{
  int speed;
  double weight;
} mycar;

int main()
{
  mycar.speed = 10;
  mycar.weight = 70;
  return 0;
}
```

```
typedef struct CAR
{
  int speed;
  double weight;
};

int main()
{
  CAR mycar;
  mycar.speed = 10;
  mycar.weight = 70;
  return 0;
}
```


made by electron & kk & rabbit125

C/C++ syntax introduction

- Iterations
  - while () → True if the statement in () is **not 0**
  - do-while() → Do the statement first then check
  - for (; ;) → for(initial ; break statement ; do)
  - switch ( ) case
  - break → Break each iterations, not if-else
  - continue → Continue each iterations, not if-else

made by electron & kk & rabbit125

C/C++ syntax introduction

- Functions
- Function data type
  - The data type of main must be **int**
- return value
  - return functions (ex: GCD)
  - return nothing for void function
- Parameters
  - Pass by value
  - Pass by reference

made by electron & kk & rabbit125

C/C++ syntax introduction

- GCD

```
int GCD( int a, int b )
{
  if (a == 0) return b;
  return GCD( b%a, a );
}
```

- For example, call GCD(14, 24)

	Call GCD(14, 24)	→	return GCD(24 % 14, 14)
→	Call GCD(10, 14)	→	return GCD(14 % 10, 10)
→	Call GCD(4, 10)	→	return GCD(10 % 4, 4)
→	Call GCD(2, 4)	→	return GCD(4 % 2, 2)
→	Call GCD(0, 2)	→	return 2

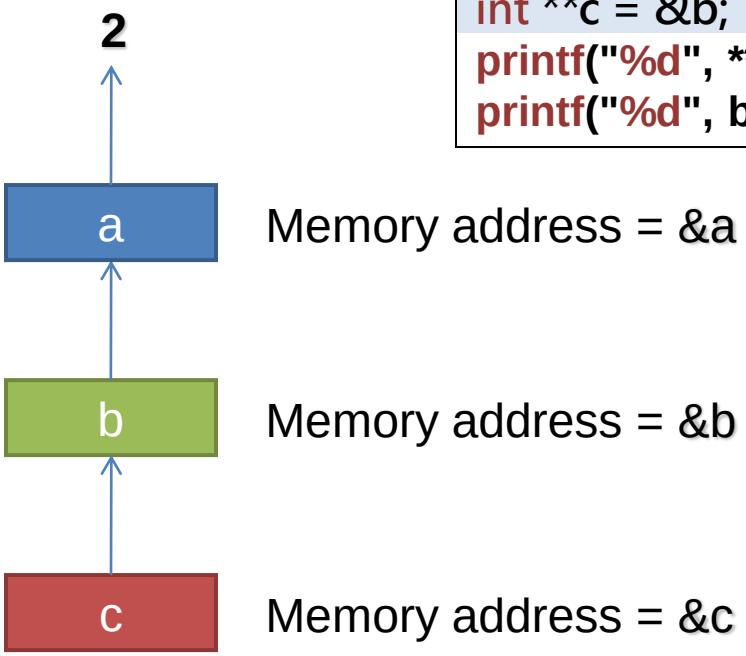
14	24
14 % 10 = 4	24 % 14 = 10
4 % 2 = 0	10 % 4 = 2

made by electron & kk & rabbit125

C/C++ syntax introduction

- Using pointer
  - Declare

```
int a = 2;
int *b = &a;
int **c = &b;
printf("%d", **c);      //2
printf("%d", b);        //Memory address of a
```



made by electron & kk & rabbit125

C/C++ syntax introduction

- Using pointer
  - Use in function

What is the difference ?

```
void swap( int *a, int *b )
{
  int tmp = *a;
  *a = *b;
  *b = tmp;
}
```

```
void swap( int a, int b )
{
  int tmp = a;
  a = b;
  b = tmp;
}
```

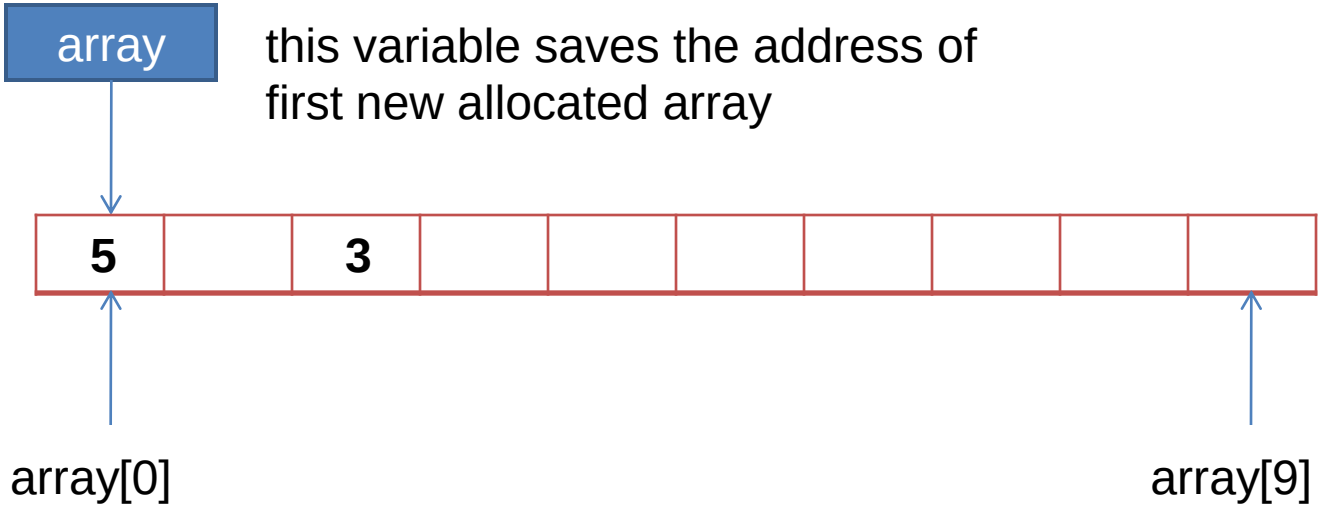

made by electron & kk & rabbit125

C/C++ syntax introduction

- Using pointer
  - Allocate array

```
int *array = (int*)malloc( 10 * sizeof(int) ) ;

array[0] = 5;
*(array + 2) = 3;
printf("%d", array[2]);        //3
```



made by electron & kk & rabbit125

C/C++ syntax introduction

- Coding style (ACM):

//headers

//define

//global variables

//functions

//main

```
#include <iostream>
#include <cstdio>
#include <cstdlib>
#include <cstring>
#include <algorithm>
using namespace std;
#define MAXN 1010

int N, K, ans;
int n[MAXN][MAXN], dp[MAXN];
bool isfind[MAXN];

int DFS(int id, int depth){
    if(id == N)return 0;
    if(isfind[id])return dp[id];
    dp[id] = 0;
    for(int i = 1;i <= n[id][0]; i++)
        dp[id] += DFS(n[id][i], depth + 1);
    isfind[id] = true;
    return dp[id];
}

int main() {
    int isover, a, b;
    while(true){
        //initial
        memset(n, 0, sizeof(n));
        memset(isfind, false, sizeof(isfind));
        //read file
        scanf("%d%d", &N, &K);
        for(int i = 1;i <= K; i++){
            scanf("%d%d", &a, &b);
            n[a][++n[a][0]] = b;
        }
    }
    return 0;
}
```


made by electron & kk & rabbit125

Outline

- C/C++ syntax introduction
- Steps to solve problems
- Time and Space Complexity Analysis
- Recursive

made by electron & kk & rabbit125

Steps to solve problems

- Step 1 : Think before you start to solve this problem !
- Step 2 : Create a new project with test file !
- Step 3 : Think what you need and includes the headers
- Step 4 : Declare the global variables you need
- Step 5 : Read the file
- Step 6 : Solve

made by electron & kk & rabbit125

Steps to solve problems

- The content of problem looks like:

內容 (Problem Description)：
迴文的定義爲正向，反向讀到的字串均相同，如：abba, abcba就是迴文。
請判斷一個字串是否是一個迴文？

輸入說明 (Input)：
一個字串(長度 < 1000)

輸出說明 (Output) ：
yes or no

First read the input and output

範例輸入 (Sample Input)：
abba
abcd

範例輸出 (Sample Output)：
yes
no

Then briefly read the sample I/O

made by electron & kk & rabbit125

Steps to solve problems

- What is palindrome?

  A *palindrome* is a word that reads the same forward as it does backward.

- How can I save a string with length no more than 1000?

  Declare a char array of size 1010. => char word[1010]

- How can I read the input file?

  Use scanf and %s to read string, such as scanf("%s", word);

- When should I stop reading the input file?

  Use while and EOF to check the End Of File, such as while(scanf("%s", word) != EOF)

- How can I check if the string is palindrome or not?

  Design a simple algorithm to check the word array

made by electron & kk & rabbit125

Steps to solve problems

- 1. We include the header file

```
#include <iostream>
#include <cstdio>
#include <cstring>
using namespace std;
```

- 2. Then declare the variable

```
#define MAXL 1010
char word[MAXL];
```

- 4. At last we design an algorithm

```
bool check(){
    int length = strlen(word);
    for(int i = 0;i < (length + 1) / 2; i++)
        if(word[i] != word[length - 1 - i])
            return false;
    return true;
}
```

- 3. Then we read the input file

```
int main() {
    while(scanf("%s", word) != EOF){
        if(check())
            printf("yes\n");
        else
            printf("no\n");
    }
    return 0;
}
```


made by electron & kk & rabbit125

Steps to solve problems

- Be careful !
  1. The size of array **cannot** be too small, otherwise you will get RE
  2. The syntax error should be avoided, otherwise you will get CE
  3. The self declared function should always on the **top of main** (Otherwise you have to declare the prototype of the function at first)
  4. Take care of infinite loop (such as `for (int i = 5;i >= 1; i++)`)
  5. Recheck you have read the input file correctly !
  6. You can design some test case yourself to recheck your program before submitting

made by electron & kk & rabbit125

Steps to solve problems

- Basic Algorithm
  - Brute Force
  - Simulation, Make Table
  - Simple Recursive Problem
- Solving Steps
  - 1. Read the Problem Description Carefully
  - 2. Induce the Sample Input and Output
  - 3. Design a Method or Algorithm to Solve it
  - 4. Use the Technical Time and Space Analysis to Check the TLE
  - 5. Write it Down
  - 6. Recheck
  - 7. Submit and Get Yes

made by electron & kk & rabbit125

FAQ & Practice

- Any questions?
- ZeroJudge a022
- NCKU OJ ID03
- NCKU OJ ID04

made by electron & kk & rabbit125

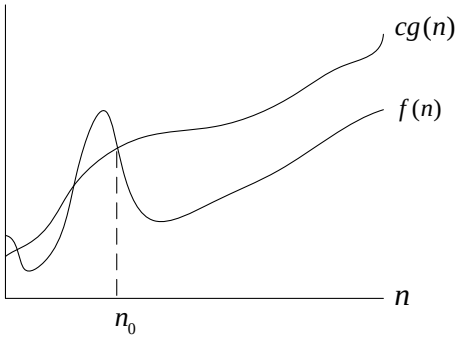
Outline

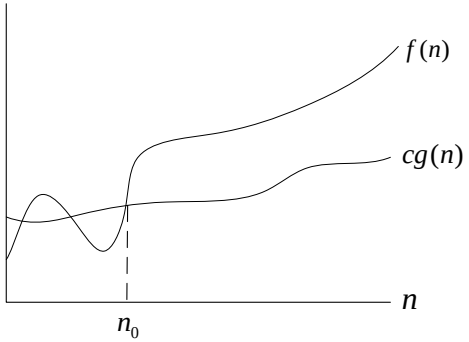
- C/C++ syntax introduction
- Steps to solve problems
- Time and Space Complexity Analysis
- Recursive

made by electron & kk & rabbit125

Time Complexity (1/3)

- Time Complexity Analysis
  - Time Limit Constraint
    - **Time Limit is the Most Important Problem During Contest!!!**
  - Use Big O as the Analysis
    - Upper Bound of the Run Time





- In this course, we only use **Technical Analysis** (non-theoretic analysis)

made by electron & kk & rabbit125

Time Complexity (2/3)

- Run Time Issue
  - Time Limit Constraint
    - **Time Limit is the Most Important Problem During the Contest!!!**
  - Technical Analysis(Input Data Size = N)
    - O(N)
      - for(int i=1; i<=N; i++) your_process_subroutine();
    - $O(N^2)$
      - for(int i=1; i<=N; i++)
        for(int j=1; j<=N; j++) your_process_subroutine();
    - $O(N^3)$, $O(N^4)$…

Door Man

Time Limit: 1000MS	**Memory Limit:** 10000K
Total Submissions: 851	**Accepted:** 294

- For the Contest and the Online Judge (Input Data Size = N)
  - Timing Constraint
    - O(N) = 1000000 ~ 2000000 equals to Run Time = 1s
      - ex: Input Data = 100, Time Limit = 1s
      - Use an $O(N^3) = O(1000000)$ Algorithm can be Used to Avoid TLE Problem

made by electron & kk & rabbit125

Time Complexity (3/3)

- Example 1:
  - Given a Problem with Data Size = 1000, Time Limit = 1s
    - Could a O(N) Algorithm Pass the Time Limited Constraint?
    - What about the O(NlogN)?
    - What about the $O(N^2)$?
    - What about the $O(N^2 \log N)$?
    - What about the O(1) time ?
- Example 2:
  - Give a Problem with Data Size = 1000000, Time Limit = 1s
    - Could a O(N) Algorithm Pass the Time Limited Constraint?
    - What about the O(logN) ?
    - What about the O(logN logN) ?
    - What about the $O(N^2)$?

made by electron & kk & rabbit125

Space Complexity (1/2)

- Space Complexity Analysis
  - **Also Use Technical Analysis**
  - Size of the Declared Array
    - int array[MAXSIZE]
      - MAXSIZE should be 1000000~2000000 (safely)
      - 4000000 will be very dangerous and limit bound
  - Local and Global
    - Local can be at most 500000
    - Global can be at most the MAXSIZE mentioned above
  - For the Contest and the Online Judge
    - Use int declaration Type for basic Analysis
    - Size of int = 4 bytes

made by electron & kk & rabbit125

Space Complexity (2/2)

- Example:
  - Local Declaration:

```
#define MAXSIZE 500000
void subroutine_1()
{
        int array[MAXSIZE];
}
```

  - Global Declaration:

```
#define MAXSIZE 1000000
int array[MAXSIZE];
int main()
{
…
}
```


made by electron & kk & rabbit125

Outline

- C/C++ syntax introduction
- Steps to solve problems
- Time and Space Complexity Analysis
- Recursive

made by electron & kk & rabbit125

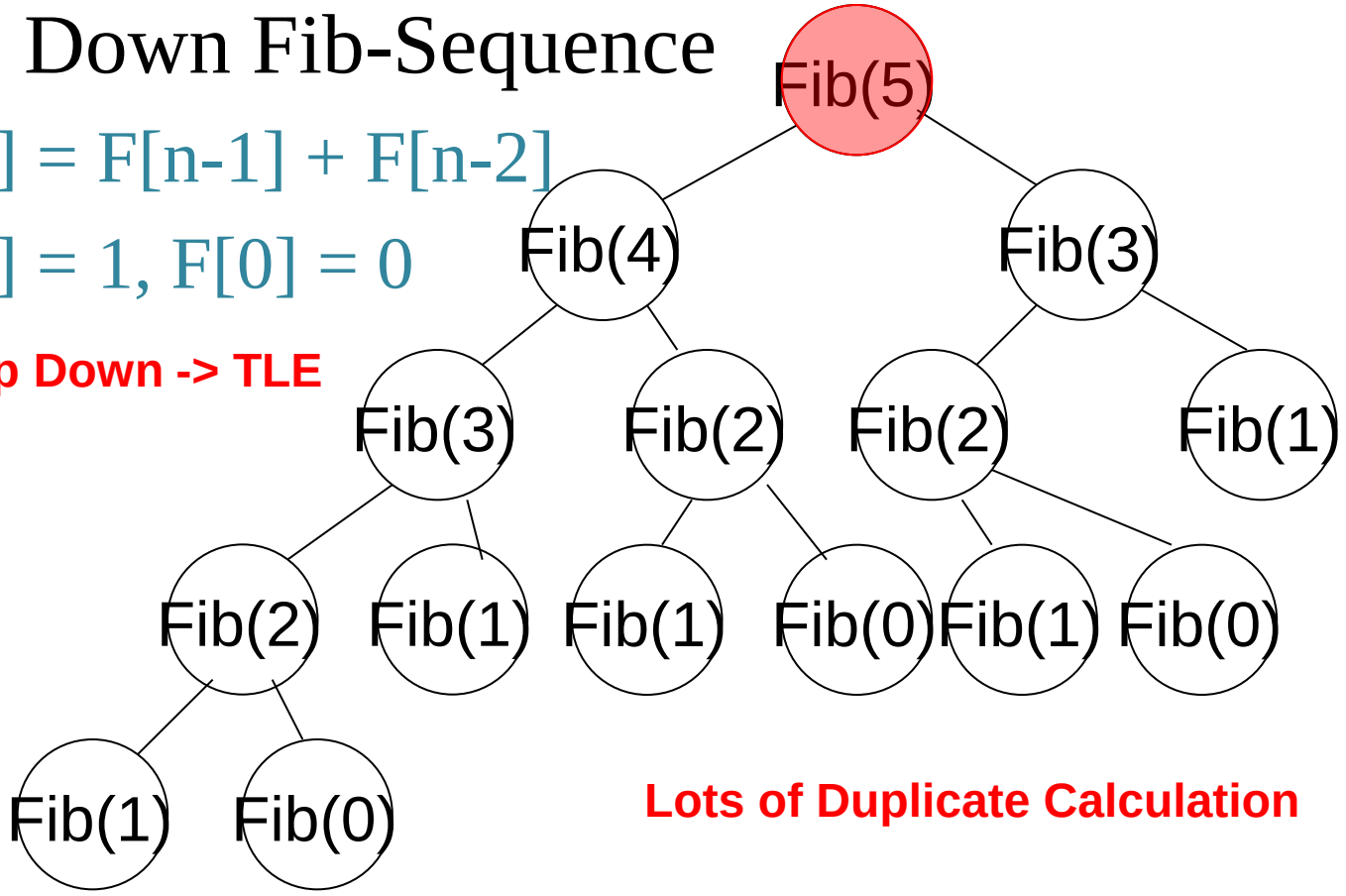
Recursive

- Top Down Fib-Sequence

  $F[n] = F[n-1] + F[n-2]$

  $F[1] = 1, F[0] = 0$

Simple Top Down -> TLE



Lots of Duplicate Calculation

made by electron & kk & rabbit125

Recursive

- Problems
  - F(30) will TLE
  - Duplicate calculation
- Solve by Top-Down's Weapon
  - **Four Steps**
    - **(1) Basic and Valid Condition**
    - **(2) Is Found Condition**
    - **(3) Recursive Part**
    - **(4) Return Value Part**

```
int fib_top_down(int n)
{
        if(n<=2) return 1;
        return fib_top_down(n-1)+fib_top_down(n-2);
}
```

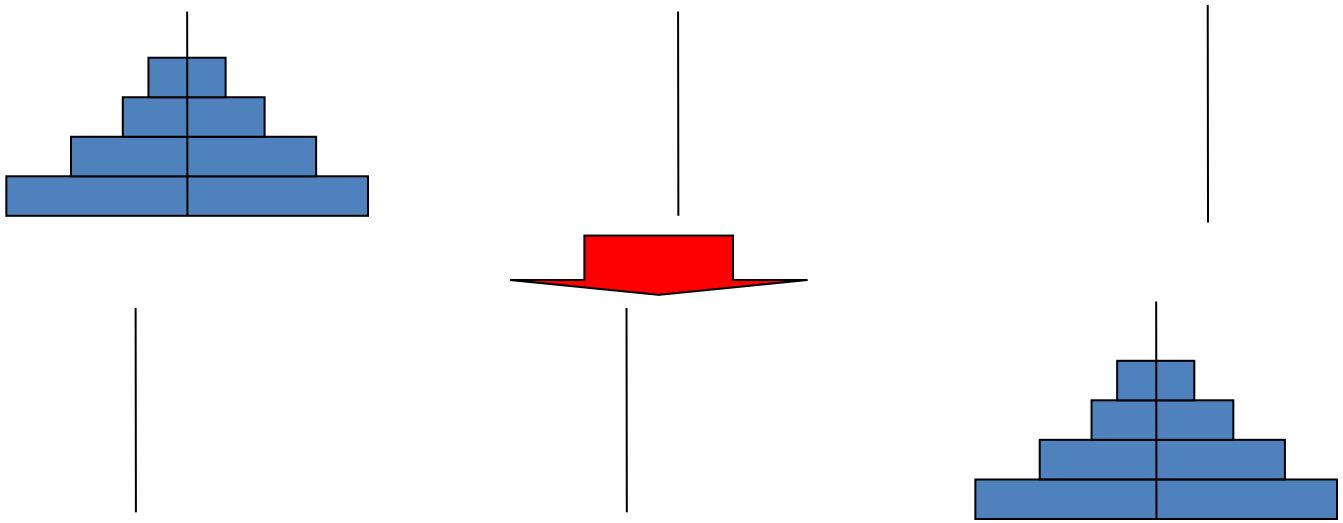
```
bool isfind[MAXN];   //all initialized to be false
int fib[MAXN];       //all initialized to be zero

int trace(int n)
{
        if(n<=2) return 1;
        if(isfind[n]) return fib[n];
        fib[n] = trace(n-1) + trace(n-2);
        isfind[n] = true;
        return fib[n];
}
```


made by electron & kk & rabbit125

Recursive

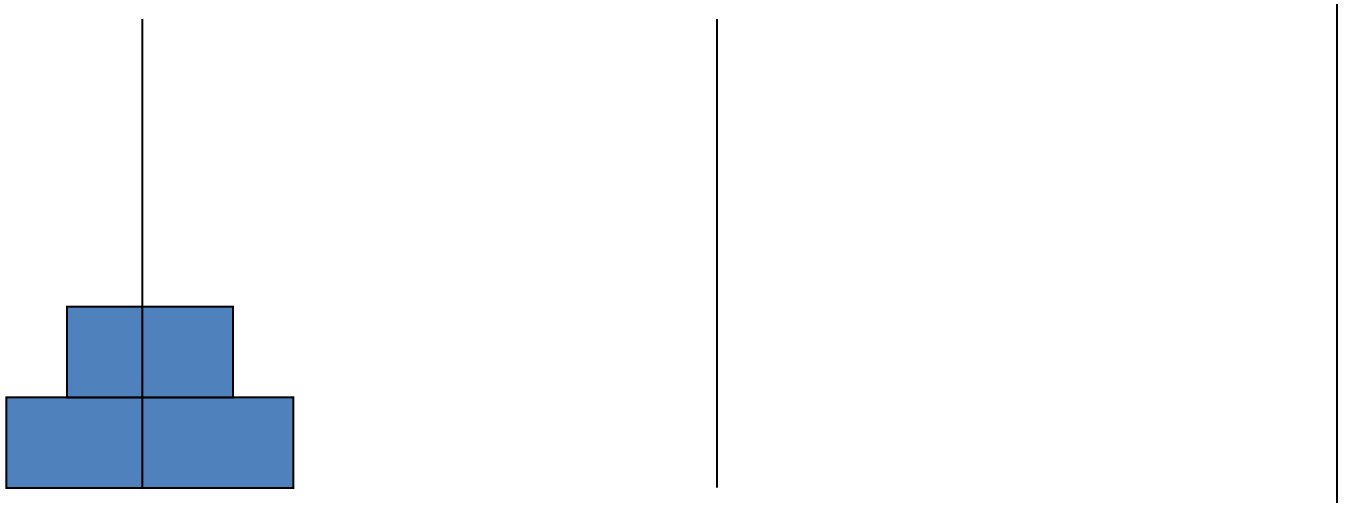
- Tower of Hanoi
  - Give three pegs, what's the number of steps required to move the tower of n blocks from the first peg to the last peg.
  - **You cannot put a large block on the small block**



made by electron & kk & rabbit125

Recursive

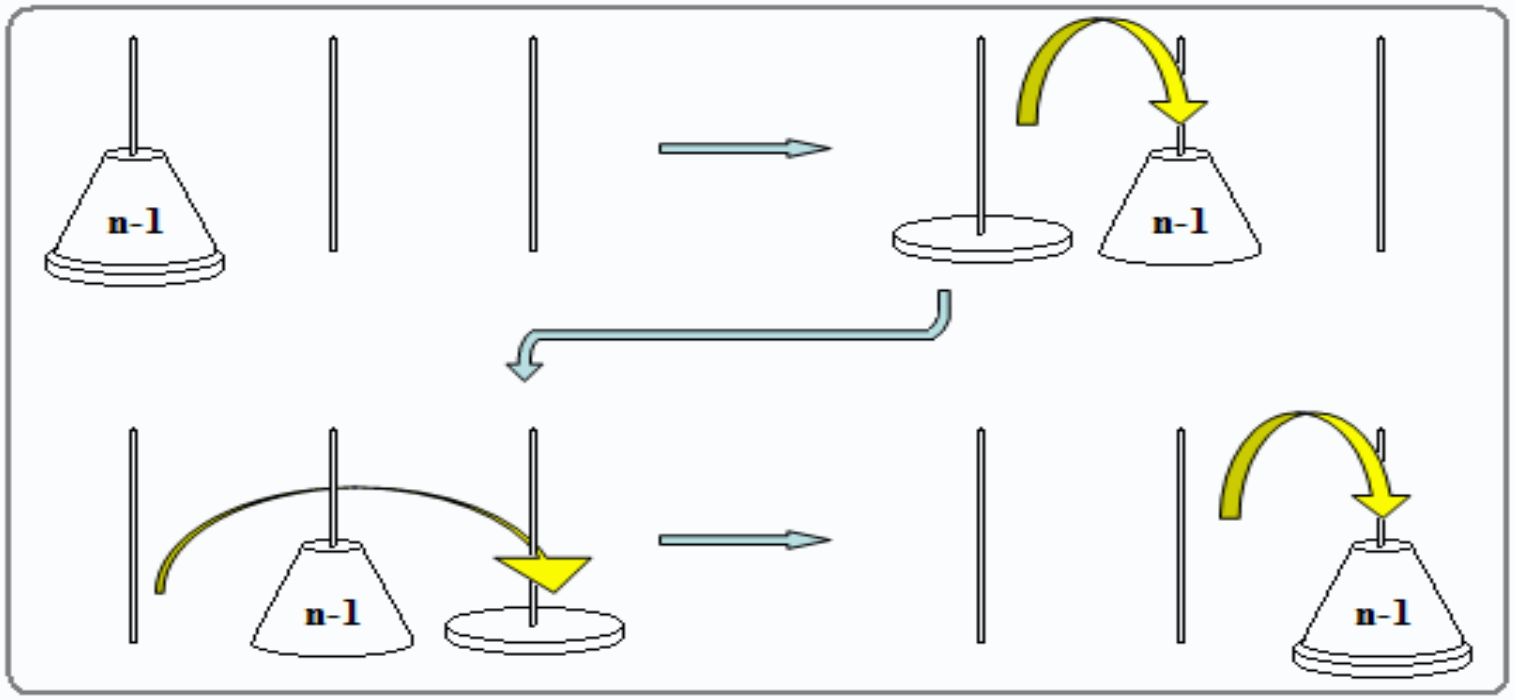
- Illustration of Tower of Hanoi



Example with a 2-block tower.

made by electron & kk & rabbit125

Recursive



Tower of Hanoi
Counting the movements

f(n) = f(n-1) + 1 + f(n-1)

made by electron & kk & rabbit125

Example

- Description
  - 把M個同樣的蘋果放在N個同樣的盤子裡，允許有的盤子空著不放，問共有多少種不同的分法？（用K表示）5，1，1和1，5，1是同一種分法。
- Input
  - 第一行是測試資料的數目t（0 <= t <= 20）。以下每行均包含二個整數M和N，以空格分開。1<=M，N<=10。
- Output
  - 對輸入的每組資料M和N，用一行輸出相應的K。
- Sample Input

  1

  7 3
- Sample Output

  8

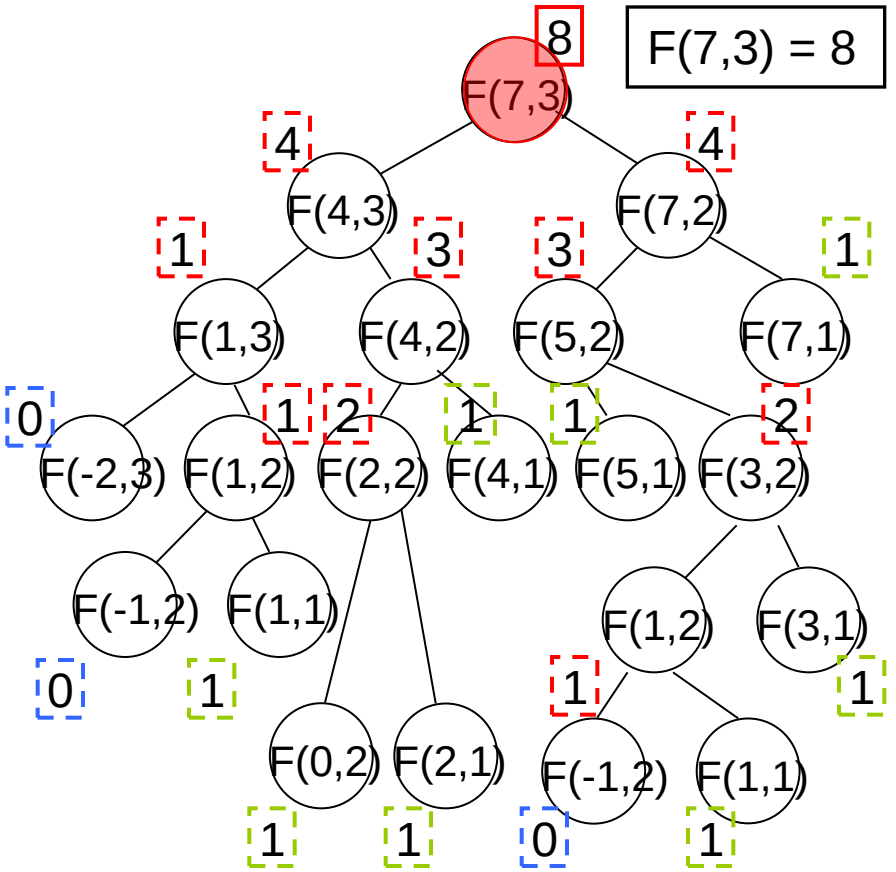
made by electron & kk & rabbit125

FAQ & Practice

- Any questions?
- POJ 1664

HINT

```
int F(int m, int n)
{
    if(m<0) return 0;
    if(m==0||n==1) return 1;
    return f(m-n,n)+f(m,n-1);
}
```



made by electron & kk & rabbit125

Homework 01

- PKU Online Judge
  - 1000 (http://acm.pku.edu.cn/JudgeOnline/problem?id=1000)
  - 1207 (http://acm.pku.edu.cn/JudgeOnline/problem?id=1207)
  - 1250 (http://acm.pku.edu.cn/JudgeOnline/problem?id=1250)
  - 3673 (http://acm.pku.edu.cn/JudgeOnline/problem?id=3673)
- UVA Online Judge
  - 10055, 10696, 944, 10499, 10323, 10300, 10302, 10346
- Zero Judge 2
  - d021 (http://140.122.185.166/ZeroJudge/ShowProblem?problemid=d021)
  - d040 (http://140.122.185.166/ZeroJudge/ShowProblem?problemid=d040)
  - p001 (http://140.122.185.166/ZeroJudge/ShowProblem?problemid=p001)
- NCKU Judge
  - Basic: 1, 2, 3 4 5, 6, 9
    11, 12, 13, 14, 15, 16, 19
  - Recursive: 25, 28, 29

made by electron & kk & rabbit125

Notice

- 多利用JUDGE討論區，BBS精華文章
- 寒假課程、線上競賽
- 大資盃結束後(1/31之後~) 或 1/23~27

made by electron & kk & rabbit125

Requirements

- Finish them by self-motion (one per day)!
- Don’t download the code from the Internet!
- Think, Create, and Solve!
- 人在做天在看
- 機會是留給有準備的人
- 加油
- Call me if need: 尤聖棨（rabbit125）
  - msn: jay_s6215@hotmail.com.tw

made by electron & kk & rabbit125