

Recursion

전자 전기 프로그래밍 실습
민 형 복 교수

For copyright and license information,
<http://class.icc.skku.ac.kr/~min/program/license.html>

Copyright (c) 2013-2014 by Hyoung Bok Min, All rights reserved.

Recursion

- Two ways of solving a problem
- (e.g.) Sum of 1 thru n, where $n \geq 1$

- Iteration

$$S_n = 1 + 2 + 3 + \dots + n-1 + n$$

- Recursion

$$S_n = \begin{cases} S_{n-1} + n, & n > 1, \text{ and} \\ 1, & n = 1 \end{cases}$$

Note that S_n call itself with a smaller parameter.

Iterative Program

```
int sum (int n)
{
    int k, total=0;
    for (k = 1 ; k <= n ; k++)
        total += k;
    return total;
}
```

Recursive Program

```
int sum (int n)
{
    int total;
    // trivial case
    if (n <= 1)
        return 1;

    // recursive expression
    total = n + sum(n-1);
    return total;
}
```

Examples

- Fibonacci sequence
- Tower of Hanoi Problem
- Quick Sort
- Binary Search
- Merge Sort
- Depth First Search
- PHP, Hypertext Preprocessor
- WINE Is Not an Emulator

Problem 1

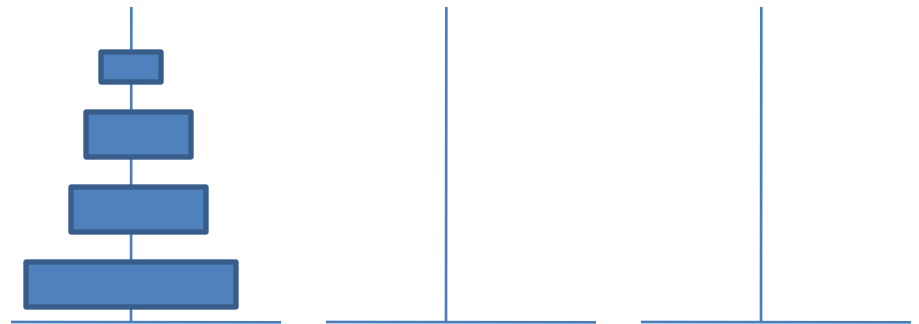
- Given an integer "n" ≥ 1 ,
- Compute factorial of n by using an iteration and a recursion.
- `int iterativeFactorial(int n);`
- `int recursiveFactorial(int n);`

Problem 2

- Given a string, print a reversed string
 - Given "I am Min, Hyong Bok"
 - Prints "koB gnuoyH ,niM ma I"
- `void reverse(char *string);`
- String의 길이를 세거나, string의 끝으로 가기 위한, 혹은 이와 유사한 표현을 쓸 수 없습니다. (e.g. `strlen()`, or similar.) 이런 표현을 쓴다면, 진정한 recursion이 아닙니다.

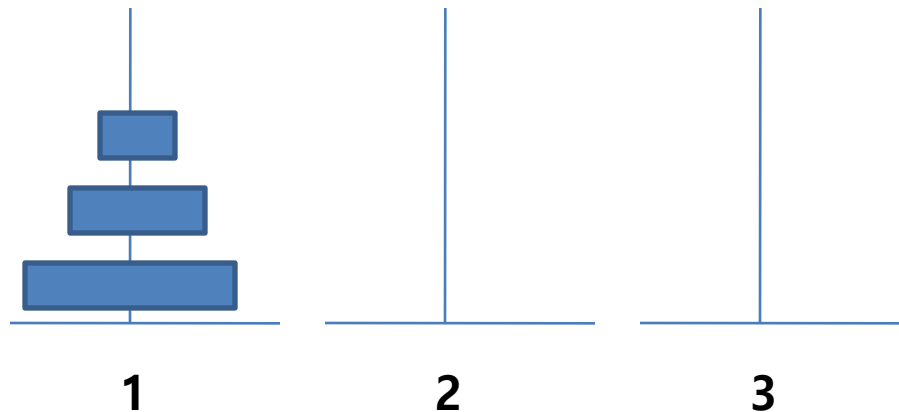
Problem 3 – Tower of Hanoi

- 옛날 옛적, Hanoi의 한 사원에 3개의 같은 탑 1, 2, 3이 있었다
- 탑1은 크기가 모두 다른 64개의 금으로 만든 donut 모양의 원판이 쌓여있었다. 큰 것일수록 아래쪽에...
- 탑2,3은 비어있다
- Hanoi의 승려들은 한 해에 한번 금 원판 하나를 옮긴다. 탑1에서 탑3으로 모든 원판을 옮기는 것이 목표이다.
- 항상, 큰 원판은 작은 원판의 아래에 있어야 한다
- 원판은 3개의 탑 이외의 장소에 둘 수 없다
- 모든 원판이 탑3으로 옮겨가는 해에 세상의 종말이 온다고 한다.



Tower of Hanoi

어떻게 옮겨야 하나?



정답

- $1 \Rightarrow 3$
- $1 \Rightarrow 2$
- $3 \Rightarrow 2$
- $1 \Rightarrow 3$
- $2 \Rightarrow 1$
- $2 \Rightarrow 3$
- $1 \Rightarrow 3$

Tower of Hanoi (code)

```
int hanoi ( int ndisks, int from, int to, int via,  
           int *from_arr, int *to_array, int size) {  
    // trivial part  
    if ( ndisks == 1 ) {  
        // record tower numbers into from_arr and to_array  
        // printf("%d -> %d\n", from, to);  
        return ??;  
    }  
    // Recursive expression  
    // move "ndisks-1" disks from 'from' to 'via'  
    // move 1 disk from 'from' to 'to'  
    // move "ndisks-1" disks from 'via' to 'to'  
    return ??;  
}
```

참고 사항

- Call the function
 `hanoi(64, 1, 3, 2, from_array, to_array, size);`
- Function returns the number of disk moves
- 필요하다면, function prototype을 변경해도 됩니다. 단, `from_array`와 `to_array`에 disk 이동 정보를 기록해야 합니다.
- 세상의 종말이 오려면 몇 년 걸리나?

$$Y_n = Y_{n-1} + 1 + Y_{n-1}, \text{ if } n > 1$$

$$= 1, \text{ if } n = 1$$

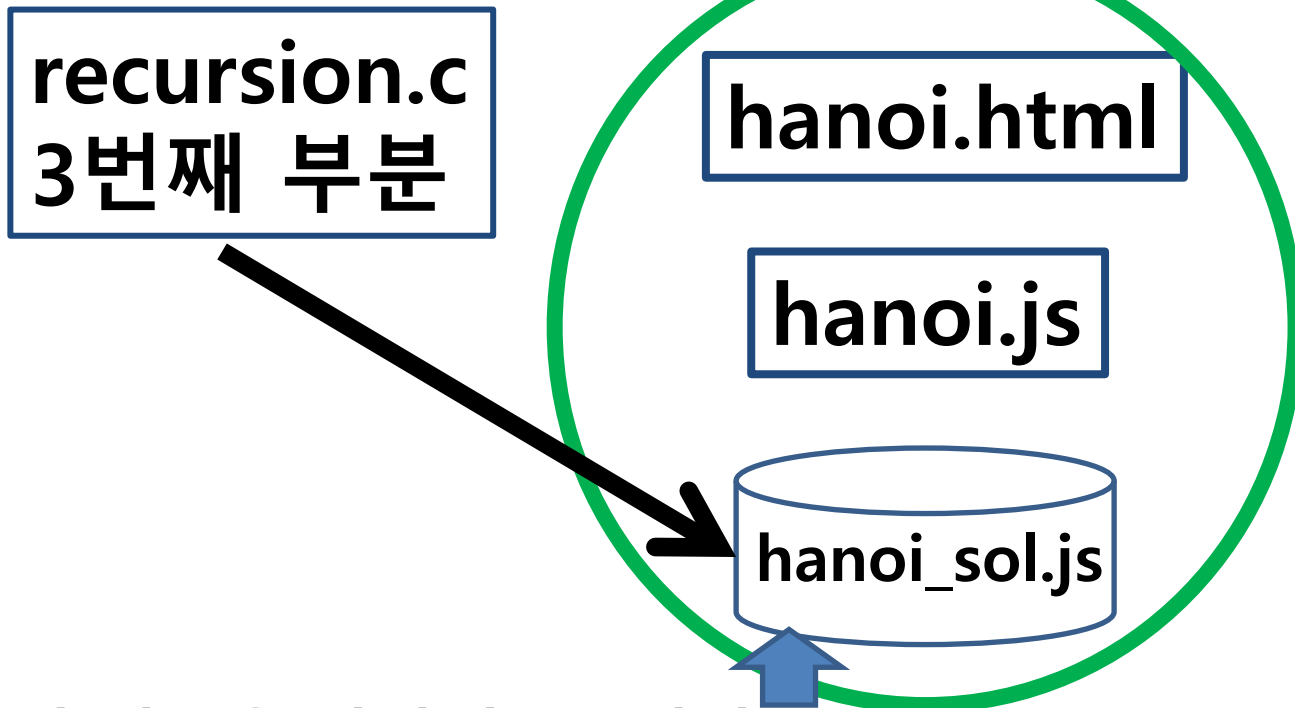
 Solution : $Y_n = 2^n - 1$

실습

- "recursion.c" is given,
- Following functions must be written at lab hours
 - int iterativeFactorial (int n)
 - int recursiveFactorial (int n)
 - void reverse (char *string)
 - int hanoi (int ndisks, int from, int to, int via, int *farr, int *tarr, int size);
Returns the number of disk moves

3번째 주제, Tower of Hanoi

Web App to show results



Disk의 이동을 나타내는 2개의 data array



Grace Hopper

(1906~1992)

[Grace Brewster Murray Hopper](#)

was one of first programmers of Havard Mark I in 1944.

She invented [the first compiler A0 system](#) on UNIVAC in 1952.

Hopper retired from U.S. Navy as rear admiral.

The picture on the left comes from [quora.com](#), and license is unknown.