


WEEK04. 재귀함수, 이진탐색

자료구조 실습

조교 소개

- ▶ 민진우
 - ▶ 컴퓨터공학과 대학원
 - ▶ 생물정보학 연구실(A1304)
 - ▶ 이메일 : mzinoo@hallym.ac.kr
- 

재귀함수; 팩토리얼; 실습01

- ▶ 재귀함수란 자기 자신을 호출하는 함수.

```
public class Main {  
    public static void main(String[] args) {  
        System.out.println(factorial(10));  
    }  
  
    public static int factorial(int n) {  
        //n은 음이 아닌 정수.  
        if(n<=1)  
            return 1;  
        else  
            return n*factorial(n-1);  
    }  
}
```

- ▶ 해당 코드를 실행해보고 결과 확인.

재귀함수; 팩토리얼; 실습02

- ▶ 재귀를 반복문으로 수정.

```
public class Main {  
    public static void main(String[] args) {  
        System.out.println(factorial(10));  
    }  
  
    public static int factorial(int n) {  
        int tmp = 1;  
        for(int i=1; i<=n; i++)  
            tmp = tmp*i;  
  
        return tmp;  
    }  
}
```

- ▶ 해당 코드를 실행해보고 결과 확인.

이원 탐색

- ▶ 정의 : 주어진 탐색키 key 가 저장된 위치(인덱스)를 찾아내는 방법.
서로 다른 n 개의 키 값을 가진 순서를 가지는 배열.
 - ▶ $key = a[mid]$ // 탐색 성공, return mid
 - ▶ $key < a[mid]$ // $a[mid]$ 의 왼편에 대해 이진 탐색.
 - ▶ $key > a[mid]$ // $a[mid]$ 의 오른편에 대해 이진 탐색.

이원 탐색; 문제01

▶ 이원 탐색의 순환 함수 표현

```
binsearch(a[], key, left, right)
// a[mid] = key인 인덱스 mid를 반환
if (left ≤ right) then {
    mid ← (left + right) / 2;
    case {
        key = a[mid] : return (mid);
        key < a[mid] : return (binsearch(a, key, left, mid - 1));
        key > a[mid] : return (binsearch(a, key, mid + 1, right));
    }
}
else return -1; // key 값이 존재하지 않음
end binsearch()
```

▶ 해당 의사코드를 보고 이원 탐색을 구현하세요.

```
public class Main {
    public static void main(String[] args) {
        int[] a = {1, 6, 13, 41, 45, 68, 70, 74, 81, 100};
        System.out.println(search(a, 81));
    }

    private static int binarySearch(int a[], int key, int left, int right) {

        코드 작성

    }

    public static int search(int a[], int key) {
        return binarySearch(a, key, 0, a.length-1);
    }
}
```

피보나치 수열

- ▶ 이 수열의 각 항은 바로 직전 두 항의 합으로 만들어짐.

- ▶ 피보나치 수열의 순환 정의

$$F_n = \begin{cases} 0 & n = 0 \\ 1 & n = 1 \\ F_{n-1} + F_{n-2} & n \geq 2 \end{cases}$$

- ▶ 결과

| n | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|-------|---|---|---|---|---|---|---|----|----|
| Value | 0 | 1 | 1 | 2 | 3 | 5 | 8 | 13 | 21 |

피보나치 수열; 문제02

- ▶ 이 수열의 각 항은 바로 직전 두 항의 합으로 만들어짐.

- ▶ 피보나치 수열의 순환 함수 표현

```
fib(n)
  if (n ≤ 0) then return 0;
  else if (n=1) then return 1
  else return (fib(n-1) + fib(n-2));
end fib()
```

- ▶ 해당 표현을 참고하여 재귀함수를 사용한 피보나치 수열을 구현하세요.

피보나치 수열; 문제03

- ▶ 이 수열의 각 항은 바로 직전 두 항의 합으로 만들어짐.

- ▶ 피보나치 수열의 반복식 표현

```
fib_i(n)
1-2   if (n < 0) then stop; // error 발생
3-4   if (n ≤ 1) then return n;
5-6   fn2 ← 0; fn1 ← 1;
7     for (i ← 2; i ≤ n; i ← i + 1) do {
8       fn ← fn1 + fn2;
9       fn2 ← fn1;
10      fn1 ← fn;
11    }
12    return fn;
13  end fib_i()
```

- ▶ 해당 표현을 참고하여 반복문을 사용한 피보나치 수열을 구현하세요.

WRITE BINARY; 문제04

- ▶ 입력받은 수를 2진수로 출력하는 프로그램 구현

- ▶ 재귀함수를 사용하여 구현하세요.

- ▶ ex) 10을 2진수로 표현하면 **1010**

- ▶ $10 / 2 = 5$

- ▶ $10 \% 2 = 0$

- ▶ $5 / 2 = 2$

- ▶ $5 \% 2 = 1$

- ▶ $2 / 2 = 1$

- ▶ $2 \% 2 = 0$

- ▶ $1 / 2 = 0$

- ▶ $1 \% 2 = 1$

- ▶ Tip)

- ▶ 2로 더 이상 나뉘지지 않을때까지 자기 자신을 호출한 후,

- ▶ 먼저 호출한 순서대로 해당 값을 출력.

과제

- ▶ 실습 두개는 코드 따라 작성해보고 결과 확인하면 됩니다.
- ▶ 문제 네개는 의사코드를 보고 직접 코드를 작성하시면 됩니다.
각 문제의 핵심 내용이 포함되거나, 해당 기능이 작동하면 됩니다.
- ▶ 실습코드와 문제들을 전부 코딩하여 제출하세요.