Data Structure

Fall 2019

M 16:00-18:00 W 11:00-13:00

http://smart.hallym.ac.kr

Instructor: Jin Kim

010-6267-8189(033-248-2318)

jinkim@hallym.ac.kr

Office Hours:

Lab(Recursion)

Fall 2019

http://smart.hallym.ac.kr

Instructor: **Jin Kim**
010-6267-8189(033-248-2318)
jinkim@hallym.ac.kr
Office Hours:

Objectives(실습목표)

- Learn about recursive definitions.재귀의 개념을 이해한다.
  더작은 형태의 자기자신을 호출.
- Base case and the general case (가장 간단한 경우, 일반적인 경우)
- 재귀의 시간복잡도를 이해한다.
- 재귀는 스택을 사용한다.
- 재귀는 반복의 다른 형태이다.

Assignments

1. Write a java program(Fiborec.java) to implement Fibonacci function using recursion(재귀사용). Calculate the running time for f(10), f(20), f(30), f(40), f(50).

2. Write a java program(Fibiter.java) to implement Fibonacci function using iteration(반복문사용). Calculate the running time for f(10), f(20), f(30), f(40), f(50). Use below code.

```
long time1 = System.currentTimeMillis ();
long time2 = System.currentTimeMillis ();
system.out.println ( ( time2 - time1 ) / 1000.0 );
```

Assignments

3. Write a java program(factrec.java) to implement factorial function using recursion.

4. Write a java program(factiter.java) to implement factorial function using iteration.

5. Write a java program(binsrch.java) to implement binary search using recursion.

6. Write a java program(binrec.java) to implement binary search using recursion.

7. 1부터 n까지의 합을 구하는 반복, 재귀프로그램을 작성하라

Assignments(참고)

8. 10진수를 이진수로 변환하는 프로그램을 재귀로 작성하라.

```
public class DecimalToBinary {

    static void decToBin(int num, int base) {
         if (num>0) {
              decToBin(num/base, base);
              System.out.print(num%base);
        }
    }
    public static void main(String[] ar) {
        int decNo = 20;
        System.out.print("십진수 " + decNo +"를 이진수로 변환 -> ");
        decToBin(decNo, 2);
        System.out.println();
    }
}
```

1부터 n까지 합

```
public class SumRecursion {

        //재귀를 사용하여 1부터 n까지의 합 구하는 sumrec 메소드 생성
    public static int sumrec(int n) {
        if(n == 1) { //n이 1이면
            return 1; //1 리턴
        }
        int result = n + sumrec(n - 1); //식 계산
        return result; //result 리턴
    }
     // 반복문을 사용하여 계산
    public static int sumiter(int n) {
        채워보라
    }
    public static void main(String[] args) {
        int n = 4; //n 생성, 4로 초기화
        System.out.println(sumrec(n)); //결과 출력
    }
}
```

Fiborec (Fibonacci recursion)

```
public class Fiborec {
	public static int fibo(int x){
		if (x <= 1)
			return x;
		else
			return fibo(x - 2) + fibo(x - 1);

	}
	public static void main(String args[]){
		int x =10;
		long time1 = System.currentTimeMillis();//지금 현재시간
		fibo(x);//피보나치 수열 돌리기
		long time2 = System.currentTimeMillis();//피보나치수열계산후 현재 시간 출력
		System.out.println((time2 - time1)/1000.0);//피보나치수열 계산 전후 시간 차이 계산
	}

}
```

Binrec(binary search recursion)

```
public class Binsrch {

	public static int Bin(int a[], int key, int min, int max) {
		if (min <= max) {
			int mid = (min + max) / 2;
			if (key == a[mid])
				return mid;
			else if (key < a[mid])
				return (Bin(a, key, min, mid - 1));
			else if (key > a[mid])
				return (Bin(a, key, mid + 1, max));
			else {
				return -1;
			}

		}
		return -1;
	}
	public static void main(String[] args) {
		int A[] = { 2, 4, 6, 8, 10, 12, 14, 16, 18, 20 };
		int in = 2;
		System.out.println("찾을 값은" + in);
		int index = Bin(A, in, 0, 10);

		if (index < 0) {
			System.out.println("해당되는 값을 찾을수 없음" + in);

		} else
			System.out.println("배열의 index값은" + index + " " + "값은" + A[index]);
	}
}
```

- * Zip all your programs(name.zip) and upload to smart.hallym.ac.kr