5645640 제갈건 0314

1. 실행코드

#include<stdio.h>

#include<time.h>

int fib\_R(int n, int a[]) { // recursion을 이용한 피보나치 수열

a[n] += 1; // 메인 함수 배열을 이용하여 순환함수의 호출횟수를 계산한다.

if (n == 0) return 0;

if (n == 1) return 1;

return (fib\_R(n - 1, a) + fib\_R(n - 2, a));

}

int fib\_i(int n) { //for문을 이용한 피보나치 수열

clock\_t start = clock(); // 함수 시작 시간 저장

if (n == 0) return 0;

if (n == 1) return 1;

int pp = 0, p = 1, result = 0;

for (int i = 2; i <= n; i++) {

result = p + pp;

pp = p;

p = result;

}

clock\_t fin = clock(); // 함수 종료시간 저장

printf("fin\_I : %d, %lf", result, (double)(fin - start) / CLOCKS\_PER\_SEC);

// 순환 함수와는 달리 해당 함수 내에서 함수 수행 시간을 바로 출력

return result;

}

int main(void) {

int k[100] = {}, n; // k[100] : 순환함수의 호출 횟수 저장, n : 함수 호출 횟수

scanf("%d", &n); // 함수 호출 횟수 입력

clock\_t start = clock(); // 계산 시작 시간 저장

int result = fib\_R(n, k); // 순환함수 실행

clock\_t fin = clock(); // 계산 종료 시간 저장

for (int i = n; i >= 1; i--) { // for문을 이용하여 순환함수 호출 횟수 출력

printf("fib\_R(%d) : %d\n", i, k[i]);

}

printf("fin\_R : %d, %lf\n", result, (double)(fin - start) / CLOCKS\_PER\_SEC); // 결과값과 수행 시산 출력

int j = fib\_i(n); // iteration을 이용한 피보나치 수열 함수

return 0;

}

1. 실행화면

텍스트이(가) 표시된 사진

자동 생성된 설명

1. 고찰

피보나치 함수의 호출 횟수를 계산하는 것에 어려움이 있었습니다. Recursion함수에서 배열을 선언하면 함수가 호출될 때마다 배열이 초기화 되므로 배열은 main함수에서 선언하고 Recursion함수 내부에서는 호출될 때마다 그에 해당하는 Array위치에 +1을 합니다.

이후 for문을 이용하여 각 함수의 호출 횟수를 출력합니다. 추가로 Recursion방식과 Iteration방식을 비교하기 위해 각 함수의 실행 시간을 출력합니다.