

PL Assignment #1-1 : Make Recursion 보고서

컴퓨터공학과 3학년

201504280 신윤호

1. 문제 해결 방법

1) sum(n)

- $n < 1$ 인 경우 0을 반환한다. 이는 순환호출의 종료조건이 된다.
- 그렇지 않은 경우 $n + \text{sum}(n-1)$ 을 반환한다. 순환호출은 $n < 1$ 인 경우까지 진행된다. $n < 1$ 이 되면 0을 반환하며 복귀를 진행한다. 매 단계마다 'n+종료된 함수에서 return된 값'이 계산된다. 가령 n의 값을 3으로 한다면 $n < 1$ 까지 순환호출을 수행한 후, 복귀 과정을 거치며 $0 \rightarrow 1+0 \rightarrow 2+1 \rightarrow 3+3$ 이 되어 최종적으로 6이 출력된다. 복귀 과정을 끝내고 맨 처음 호출된 함수까지 종료되면 1부터 n까지의 합계가 반환한다.

2) fibonacci(n)

- 피보나치 수열은 순환적으로 정의할 수 있으므로 이에 따라 구현한다.
- $n == 0$ 인 경우 0을 반환하고, $n == 1$ 인 경우 1을 반환한다. 이 둘은 순환호출의 종료조건이 된다.
- 그 외의 경우 $\text{fibonacci}(n-1) + \text{fibonacci}(n-2)$ 를 반환한다. 즉, $n-1$ 과 $n-2$ 에 대해 각각 순환호출을 수행한 후 그 값을 더해 반환한다. 순환호출은 $n == 0$ 이거나 $n == 1$ 인 시점까지 $n-1$ 과 $n-2$ 에 대해 각각 수행되므로 마치 이진 트리와 같은 형태로 순환호출을 표현할 수 있다. 종료조건에 도달하면 $n == 0$ 인 경우 0을 반환, $n == 1$ 인 경우 1을 반환하며 함수가 복귀를 진행한다. 복귀의 매 단계마다 이전 단계에서 반환된 두 값을 더한 값을 반환하게 되며 맨 처음 호출한 함수가 종료되면 피보나치 수열의 n번째 항의 값을 반환한다.

3) factorial(n)

- 팩토리얼은 순환적으로 정의할 수 있으므로 이에 따라 구현한다.
- $n \leq 1$ 인 경우 1을 반환한다. 이는 순환호출의 종료조건이 된다.
- 그 외의 경우 $n * \text{factorial}(n-1)$ 을 반환한다. 순환호출은 $n \leq 1$ 이 될 때까지 진행되고 $n \leq 1$ 인 단계에서 1을 반환하며 복귀를 진행한다. 매 단계마다 'n*종료된 함수에서 return된 값'이 계산되어 맨 처음 호출된 함수까지 종료되면 팩토리얼 계산 값을 반환한다.

4) decimal_to_binary(n)

- $n < 2$ 인 경우 n을 출력한다. 이는 순환호출의 종료조건이 된다.
- 그 외의 경우 $(10 * \text{decimal_to_binary}(n/2)) + n \% 2$ 를 반환한다. 즉, $n/2$ 을 인자로 전달해 함수를 순환호출해 받은 return값에 10을 곱해준 후, n을2로 나눈 나머지를 더해 반환한다. 10을 곱해주는 이유는 자리수를 정해주기 위함이다. 최종적으로 $n < 2$ 인 경우까지 순환호출 된다. $n < 2$ 인 단계에서 n을 반환한 후, 함수 종료와 함께 복귀를 진행하며 $(10 * \text{decimal_to_binary}(n/2)) + n \% 2$ 문장이 실행된다. 맨 처음 호출된 함수까지 종료되면 전체 과정이 종료된다.

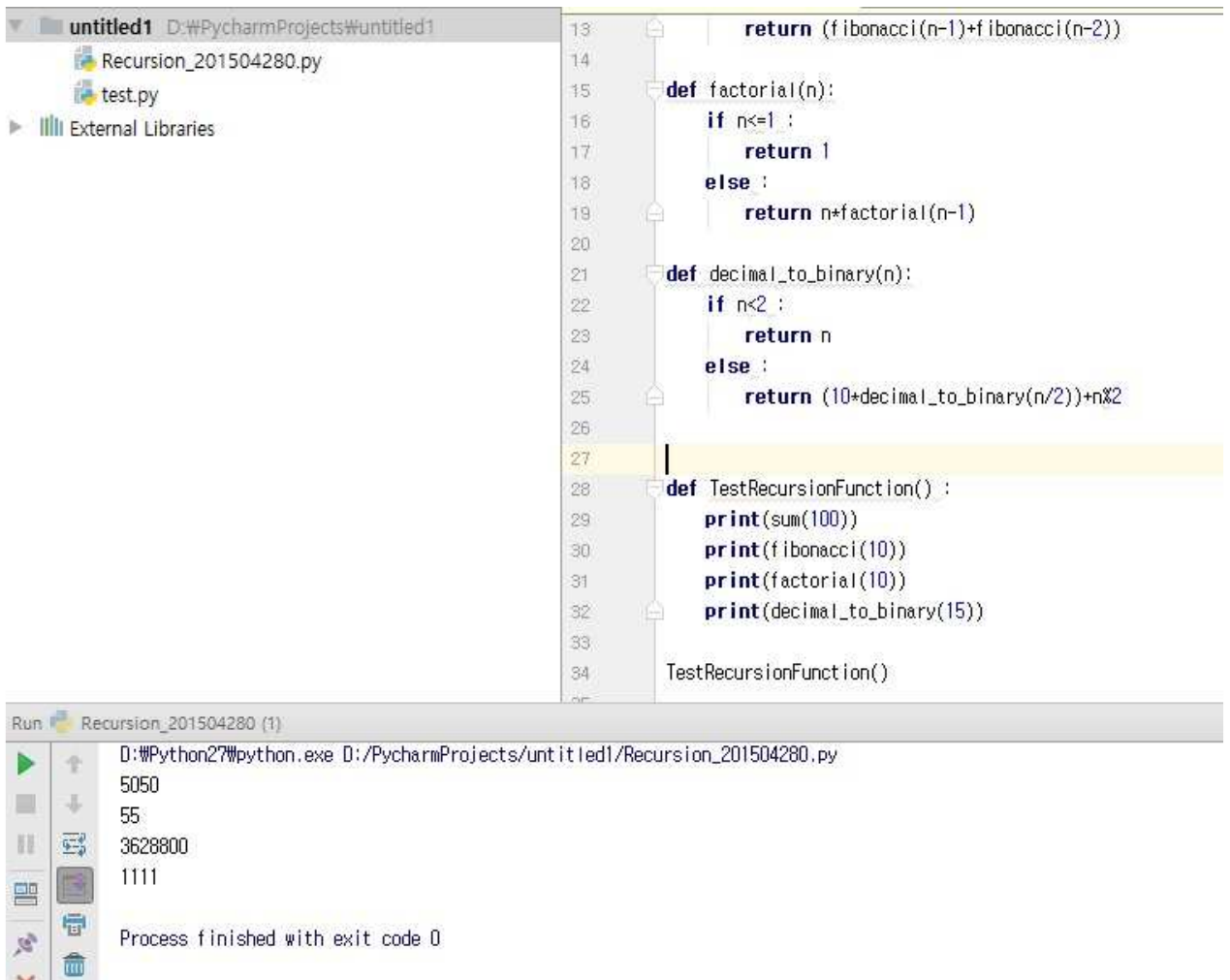
2. 느낀점

자료구조 과목에서 배웠던 순환을 다시 상기할 수 있는 좋은 기회가 되었던 것 같다. 또한

파이썬은 처음 사용해보았는데, 비록 간단한 코드지만 좋은 연습이 된 것 같고 파이썬이라는 언어에 대한 흥미가 생겨나는 것 같다. 특히 자료형을 명시하지 않아도 되는 점, 함수나 if문을 작성할 때 중괄호를 사용하지 않는 점 등이 눈에 띄었다. 뿐만 아니라 10진수를 2진수로 변환하는 알고리즘은 이번에 처음 접해본 알고리즘이어서 좋은 공부가 되었던 것 같다.

3. 테스트 코드 실행 결과

```
def TestRecursionFunction() :  
    print(sum(100))  
    print(fibonacci(10))  
    print(factorial(10))  
    print(decimal_to_binary(15))  
TestRecursionFunction()
```



The screenshot shows the PyCharm IDE interface. The top pane displays the source code for 'Recursion_201504280.py'. The code defines three recursive functions: `fibonacci`, `factorial`, and `decimal_to_binary`. The `TestRecursionFunction` function calls these functions and prints their results. The bottom pane shows the output of the program execution, which matches the expected results from the code.

```
13     return (fibonacci(n-1)+fibonacci(n-2))  
14  
15     def factorial(n):  
16         if n<=1 :  
17             return 1  
18         else :  
19             return n*factorial(n-1)  
20  
21     def decimal_to_binary(n):  
22         if n<2 :  
23             return n  
24         else :  
25             return (10+decimal_to_binary(n/2))+n%2  
26  
27  
28     def TestRecursionFunction() :  
29         print(sum(100))  
30         print(fibonacci(10))  
31         print(factorial(10))  
32         print(decimal_to_binary(15))  
33  
34     TestRecursionFunction()
```

Run Recursion_201504280 (1)
D:\Python27\python.exe D:/PycharmProjects/untitled1/Recursion_201504280.py
5050
55
3628800
1111
Process finished with exit code 0