

자료구조 1주차 과제

제출일: 2022. 9. 8

이름/학번: 윤준영 / 2016121150

문1) 다음 프로그램에 대한 설명이다. 이에 대하여 참인지, 거짓인지를 쓰고 근거를 설명하시오.(3)

```
main() {  
    int x[1000]  
    int i;  
    for (i=0; i<1000; i++)  
        x[i] = i;  
    print("%d", foo1(x, 1000));  
}
```

```
int foo1(int a[], int n) {  
    int i, j;  
    int s = 0;  
    for (i=1; i<=n; i=i+1) {  
        for (j=0; j<i; j++)  
            s+=a[j];  
    }  
    return s;  
}
```

$0 \sim 999 \rightarrow x$

0	1	2	...	999
---	---	---	-----	-----

$i = 1$

$i = 2$

$i = 3$

$(x[0]) + (x[0] + x[1]) + (x[0] + x[1] + x[2])$

$+ \dots + (x[0] + x[1] + \dots + x[n-1])$

$$T(n) = 1 + 2 + 3 + \dots + n \\ = \frac{n(n+1)}{2}$$

1) main함수는 입력이 없으므로, 알고리즘이라고 할 수 없다.

거짓 입력이 0이상, 출력이 1이상이면 알고리즘이라고 할 수 있다.

2) foo1 함수의 최선의 경우 시간 복잡도는 인자 n에 대하여 $O(n^2)$ 이다.

참 foo1함수의 연산횟수는 $1 + 2 + 3 + \dots + n$ 이고 $= \frac{n(n+1)}{2} \sim O(n^2)$ 이다.
0이상의 정수 n에 대해 모든 시간복잡도는 $O(n^2)$ 이다.

3) 배열 a의 크기를 n이라고 할 경우 foo1함수의 최악의 경우 공간 복잡도는 $O(n)$ 이다.

거짓 C언어는 배열의 포인터를 전달하므로 공간 복잡도는 $O(1)$ 이다.

문2) 다음 함수의 최악의 경우 시간 복잡도를 점근적 복잡도로 쓰고 근거를 설명하시오.(3)

1)

```
int fool(int a, int n) { /* 인자 n은 항상 양수 또는 0으로 가정함 */
    if (a == 0)
        return 0;
    else if (n <= 0)
        return 1;
    return (a * fool(a, n-1));
}
```

$$T(1) = c + T(0)$$

$O(n)$

$$T(2) = c + T(1)$$

$$= c + c + T(0)$$

$$T(3) = c + c + c + T(0)$$

$$T(n) = c + c + \dots + c + T(0)$$

if else문의 연산을 C로 놓으면

$$T(n) = \underbrace{C}_{C'} + 1 + T(n-1) = C' + T(n-1)$$

$$= C' + C' + T(n-2)$$

\vdots

$$= n \times C' \sim O(n)$$

2)

```
int foo2(int a[], int n) {
    int i;
    int s = 0;
    for (i=1; i<n; i=i*2) {
        s+=a[i];
    }
}
```

$O(\log_2 n)$

n	1	2	3	4	5	...	8	9	...	2^n	2^{n+1}	...
$T(n)$	0	1	2	2	3	...	3	4	...	n	$n+1$...

foo2함수의 계산횟수는 10(상의)인 n 에 대하여 $\log_2^{(n-1)} + 1$ 의 정수부분이다.

3) foo4 함수의 복잡도를 제시하시오.

```
int foo4(int a[], int n) {
    int i;
    int s = 0;
    for (i=1; i<n; i=i+1) {
        s += foo3(a, i);
    }
}
```

```
int foo3(int a[], int n) {
    int i;
    int s = 0;
    for (i=1; i<n; i=i+1) {
        s += a[i];
    }
    return s
}
```

$1 \sim n-1$

$O(n^2)$

foo3 함수에서 반복문을 $i=1 \sim (n-1)$, $n-1$ 번 덧셈연산이 이루어지므로 $T_3(n) = n-1 + C_3$

$$\begin{aligned} \text{foo4 함수에서 } T_4(n) &= T_3(1) + 1 + T_3(2) + 1 + \dots + T_3(n-1) + 1 + C_4 \\ &= T_3(1) + T_3(2) + \dots + T_3(n-1) + n + C_4 \\ &= 0 + 1 + \dots + n-2 + (n-1)(C_3 + C_4) \\ &= \frac{(n-2)(n-1)}{2} + nC_3 + C'' \sim O(n^2) \end{aligned}$$

문3)

```
main() {
    int x[1000]
    int i;
    int r;
    for (i=0; i<1000; i++)
        x[i] = i;

    r = fool(x, 1000);
}

int fool(int a[], int n) {
    int i, j;
    int s = 0;
    for (i=1; i<=n; i=i+1) {
        for (j=0; j<i; j++)
            s+=a[j];
    }
    return s;
}
```

1) gettimeofday 함수와 timeval 구조체를 사용하여 fool 함수 호출 수행에 소요되는 시간을 출력하도록 프로그램을 추가로 작성하시오.(실행 화면을 보이시오) (3)

```
HW1 > C hw1.c > ...
1  #include <stdio.h>
2  #include <sys/time.h>
3
4  int fool(int a[], int n) {
5      int i, j;
6      int s = 0;
7      for (i=1; i<=n; i=i+1) {
8          for (j=0; j<i; j++)
9              s+=a[j];
10     }
11     return s;
12 }
13
14 main() {
15     struct timeval tstart, tfinish;
16     double tsecs;
17     int x[1000];
18     int i;
19     int r=0;
20     for (i=0; i<1000; i++)
21         x[i] = i;
22     gettimeofday(&tstart, NULL);
23     r = fool(x, 1000);
24     gettimeofday(&tfinish, NULL);
25     tsecs = (tfinish.tv_sec-tstart.tv_sec) +
26             1e-6 * (tfinish.tv_usec-tstart.tv_usec);
27     printf("time: %f\n", tsecs);
28     printf("value: %d\n", r);
29 }
```

time : 0.000996 sec
value : 166666500

문제 1 출력 디버그 콘솔 터미널 JUPYTER

time: 0.000996
value: 166666500

* 터미널이 작업에서 다시 사용됩니다. 닫으려면 아무 키나 누르세요.

2) 1)에서 작성한 시간 측정 프로그램을 윈도우10 PC에서 수행한다고 할 때, 출력된 소요 시간이 함수 수행이 CPU를 사용한 시간을 정확히 출력한다고 할 수 있는가? 주장에 대한 근거를 제시해 보시오.(1)

정확히 출력한다고 할 수 없다.

위 1)에서 0.000996s가 걸렸지만 다시 실행해보니, 0.000995s가 걸렸고, PC로 다른 작업을 하는 중 실행해보았을 때에는 0.001995s, 0.002010s가 걸려, 약 2배의 시간이 측정되었다. 왜냐하면 윈도우 10 운영체제이기 돌아가고 있는 다른 프로그램이 CPU를 사용하기 때문이다.

따라서 출력된 시간이 무엇이 해당 함수 수행이 CPU를 사용한 시간이라고 하기 어렵다.