

**기출문제 분석**

1. <보기>에서 함수 f()의 시간복잡도는? [2022년 서울 7급]

-----<보기>-----

```
int f(int n) {
    int count = 0;
    for (int i = n; i > 0; i /= 2)
        for (int j = 0; j < i; j++)
            count += 1;
    return count;
}
```

- ①  $\Theta(n)$
- ②  $\Theta(n \log n \log n)$
- ③  $\Theta(n \log n)$
- ④  $\Theta(n^2)$

☞ 시간복잡도

// 먼저, 주어진 코드에 대한 시간복잡도가  $\Theta(n \log n)$ 이 아닌가? 질문하는 분들이 많다.  
 // 다음에 제시한 두 코드를 분석해보면, 시간복잡도가  $O(n)$ 인 것을 어느 정도 이해할 수 있다.

<pre>int f(int n) {     int count = 0;     for (int i = n; i &gt; 0; i /= 2) //log n         for (int j = 0; j &lt; i; j++)             count += 1;     return count; }</pre>	<pre>int f(int n) {     int count = 0;     for (int i = n; i &gt; 0; i /= 2) //log n         for (int j = 0; j &lt; n; j++)             count += 1;     return count; }</pre>																								
n = 16인 경우																									
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <th style="width: 10%;">i</th> <th style="width: 90%;">for(j=0; j&lt;i; j++)</th> </tr> <tr><td>16</td><td>j = 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15</td></tr> <tr><td>8</td><td>j = 0 1 2 3 4 5 6 7</td></tr> <tr><td>4</td><td>j = 0 1 2 3</td></tr> <tr><td>2</td><td>j = 0 1</td></tr> <tr><td>1</td><td>j = 0</td></tr> </table>	i	for(j=0; j<i; j++)	16	j = 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15	8	j = 0 1 2 3 4 5 6 7	4	j = 0 1 2 3	2	j = 0 1	1	j = 0	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <th style="width: 10%;">i</th> <th style="width: 90%;">for(j=0; j&lt;n; j++)</th> </tr> <tr><td>16</td><td>j = 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15</td></tr> <tr><td>8</td><td>j = 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15</td></tr> <tr><td>4</td><td>j = 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15</td></tr> <tr><td>2</td><td>j = 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15</td></tr> <tr><td>1</td><td>j = 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15</td></tr> </table>	i	for(j=0; j<n; j++)	16	j = 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15	8	j = 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15	4	j = 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15	2	j = 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15	1	j = 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
i	for(j=0; j<i; j++)																								
16	j = 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15																								
8	j = 0 1 2 3 4 5 6 7																								
4	j = 0 1 2 3																								
2	j = 0 1																								
1	j = 0																								
i	for(j=0; j<n; j++)																								
16	j = 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15																								
8	j = 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15																								
4	j = 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15																								
2	j = 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15																								
1	j = 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15																								
실행횟수 = 16 + 8 + 4 + 2 + 1 = 31	실행횟수 = 16 + 16 + 16 + 16 + 16 = 80																								
i = n인 경우, 실행횟수																									
for(j=0; j<i; j++)	for(j=0; j<n; j++)																								
$n + n/2 + n/4 + n/8 + \dots + n/n = 2n - 1 = O(n)$	$\log n \times n = O(n \log n)$																								

## // 추가 정리 - 만약, 복잡도를 엄격하게 적용한다면?

$$O(n) < ??? < O(n \log n)$$



즉, 복잡도를  
더 미세하게 분류하면  
???가 존재할 수 있다.  
단지, 복잡도를  
미세하게 분류하지 않으므로  
???는  $O(n)$ 이다.

- 시간복잡도, 즉 점근표기법을 깊이 공부하려면 공부할 것이 많다.
- 단지, 시험에서 점근표기법을 너무 깊이 공부할 필요는 없다.
- 앞에서 제시한 방식으로 점근표기법을 분석하면 쉽게 이해될 수 있다.