

알고리즘(복잡도)

이름:

학번:

1. 다음 함수 func()의 연산 횟수를 구하고 시간복잡도를 Big-O 표기로 쓰시오

```
void func(int *a, n)
{
    int i=0, j=0;
    for (i = 0 ; i < n-1 ; i++)
        for(j=i+1; j<n ; j++)
            if (a[i] == a[j]) a[j] = 0 ;
}
```

2. 다음은 배열의 합을 구하는 위한 알고리즘은 작성한 것이다. 시간 복잡도가 $\Theta(N)$ 임을 보이시오.

1. $sum \leftarrow 0$
2. **for** $i \leftarrow 1$ **to** n **do**
3. $sum \leftarrow sum + A[i]$
4. **return** sum

알고리즘(복잡도)

이름:

학번:

3. $x[1,2,3,4,\dots,n]$ 의 배열에서 가장 큰수를 찾는 알고리즘을 작성한 후 시간복잡도를 Big-O 표기로 쓰시오.

```
max ← x[0];
```

```
for i ← 1 to n do
```

```
    if x[i] > max
```

```
        then max ← x[i];
```

```
return max;
```

4. 다음 코드의 시간복잡도를 Big-O 표기로 쓰시오.

```
for (i ← 0; i < n; i ← i + 1) {  
    for (j ← 0; j < n; j ← j + 1) {  
        sum ← sum + j;  
    }  
}
```