

시간복잡도



1단계: 기본 반복문 예제

예제 1. 중첩 반복문

```
for i in range(n):    T(n) = n^2 => O(n^2)
    for j in range(n):
        print(i, j)
```



2단계: 선형 반복 + 조건문

예제 2. 2중 반복이지만 j가 i 이후로만 순회

```
for i in range(n):    T(n) = 1/2n^2 => O(n^2)
    for j in range(i, n):
        print(i, j)
```



3단계: 로그가 포함된 반복문

예제 3. 반복마다 절반씩 줄어드는 경우

```
i = n
while i > 1:    T(n) = O(log n)
    i /= 2
```



4단계: 선형 + 로그 반복문

예제 4. 반복마다 절반씩 줄어드는 내부 루프

```
for i in range(n):
    j = n
```

while $j > 0$:

$j // 2$

$$T(n) = O(n \log n)$$

🧠 5단계: 분할정복 기본형

예제 5. 이진 분할 정복

```
def f(n):  
    if n <= 1:  
        return 1  
    return f(n // 2) + f(n // 2)
```

$$T(n) = 2T\left(\frac{n}{2}\right) + O(1)$$

$$= O(n)$$

🧩 6단계: Merge Sort 유형

예제 6. 합병정렬

```
def merge_sort(arr):  
    if len(arr) <= 1:  
        return arr  
    mid = len(arr)//2  
    left = merge_sort(arr[:mid])  
    right = merge_sort(arr[mid:])  
    return merge(left, right) # O(n)
```

$$T(n) = 2T\left(\frac{n}{2}\right) + O(n)$$

$$\Rightarrow O(n \log n)$$

🧩 7단계: 분할은 여러 개지만 정복은 가벼운 경우

예제 7. 삼등분 분할 정복

```
def tri(n):  
    if n <= 1:  
        return 1  
    return tri(n//3) + tri(n//3) + tri(n//3)
```

$$T(n) = 3T\left(\frac{n}{3}\right) + O(1)$$

$$= O(n)$$

8단계: Karatsuba 곱셈형 (고급)

예제 8. 분할정복 곱셈

```
def karatsuba(x, y):
    if x < 10 or y < 10:
        return x * y
    n = max(len(str(x)), len(str(y)))
    m = n // 2
    high1, low1 = divmod(x, 10**m)
    high2, low2 = divmod(y, 10**m)
    z0 = karatsuba(low1, low2)
    z1 = karatsuba((low1 + high1), (low2 + high2))
    z2 = karatsuba(high1, high2)
    return (z2 * 10**(2*m)) + ((z1 - z2 - z0) * 10**m) + z0
```

$$T(n) = 3T\left(\frac{n}{2}\right) + O(n)$$
$$= O(n^{\log_2 3})$$

9단계: 복합형 예제

예제 9. 분할정복 + 로그 루프 결합

```
def g(n):
    if n <= 1:
        return 1
    for i in range(n):
        k = n
        while k > 1:
            k //= 2
    return g(n//2) + g(n//2)
```

$$T(n) = 2T\left(\frac{n}{2}\right) + O(n \log n)$$

by 멀티로그

$$T(n) = O(n \log n)$$