

2장, 해답

<2.1절>

2.1

- (1) 리스트의 크기, 비교연산
- (2) 리스트의 크기, 덧셈연산
- (3) n , 곱셈연산
- (4) n , 덧셈연산
- (5) a , b , 나눗셈연산
- (6) a , b , 비교연산(3행)
- (7) a , b , 나머지연산(1행 약수 찾기)

2.2

- (1) 최선, 최악, 평균 모두 동일: $O(n)$
- (2) 최선, 최악, 평균 모두 동일: $O(n)$
- (3) 최선, 최악, 평균 모두 동일: $O(n)$
- (4) 최선, 최악, 평균 모두 동일: $O(n^2)$
- (5) 최선: $O(1)$

최악(어려움): $O(\log b)$ <https://www.baeldung.com/cs/euclid-time-complexity>

- (6) 최선: $O(a+b)$, 최악/평균: $O(ab)$
- (7) 최선, 최악, 평균: $O(a)$

2.3

- (1) 증가속도: $O(n)$, 최선 $O(1)$, 최악/평균 $O(n)$
- (2) 증가속도: $O(n)$, 최선/최악/평균 $O(n)$

2.4

- (1) =
- (2) >
- (3) <
- (4) =
- (5) <

2.5

$$(1) \frac{1 + \log_2 n}{\log_2 n}$$

$$(2) 2$$

$$(3) 4$$

$$(4) 4$$

$$(5) \sqrt{2}$$

<2.2절>

2.6

$$(1) O(n^2)$$

$$(2) O(n^3)$$

$$(3) O(n^3)$$

$$(4) O(3^n)$$

$$(5) O(n!)$$

2.7

$$O(1) \quad O(\log n) \quad O(n) \quad O(n \log n) \quad O(n^2) \quad O(n^3) \quad O(2^n) \quad O(n!)$$

2.8

$$\log_2 n \quad \ln^2 n \quad 0.0001n^4 + 3n + 1 \quad 2^{3n} \quad 3^n \quad 4^n \quad (n - 10)!$$

2.9*

$$(1) c=1 \text{와 } n_0=1 \text{가 존재}$$

$$(2) c=4 \text{와 } n_0=1 \text{가 존재}$$

$$(3) 0.000001n^3 \notin O(n^2) \text{ 증명:}$$

$0.000001n^3$ 이 $O(n^2)$ 에 포함된다고 가정한다. 그러면 $n \geq n_0$ 인 모든 n 에 대해 $|0.000001n^3| \leq c|n^2|$ 을 만족하는 c 와 n_0 가 존재해야 한다. 부등식의 양변을 n^2 으로 나누면 $|0.000001n| \leq c$ 이 되고, 상수 c 를 아무리 크게 잡더라도 그보다 큰 n 이 존재할 수 있다. 따라서 위의 가정은 틀렸고, $0.000001n^3$ 는 $O(n^2)$ 에 포함될 수 없다.

$$(4) \sim (9) \text{ 같은 방법으로 증명.}$$

2.10

- 1) 동전을 $n/2$ 개씩 두 그룹으로 나눈다. 홀수인 경우 하나를 제외하고 나눈다.
- 2) 양팔저울로 측정한다.
- 3) 만약 같으면(동전이 홀수인 경우에만 해당함) 남은 동전이 불량이다. 양팔저울로 정상 동전 하나와 비교하여 무거운지 가벼운지를 판단한다.
- 4) 같지 않다면, 무거운 그룹을 다시 두 그룹으로 나누어 측정한다.
- 5) 만약 두 그룹이 같으면 불량 동전은 가볍다.
- 6) 만약 다르면 불량 동전은 무겁다.

<2.3절>

2.11

- (1) 135
- (2) 4196352
- (3) n
- (4) n^2
- (5) $1000n$
- (6) $\frac{n(n+1)}{2}$
- (7) $\frac{1}{6}n(n+1)(2n+1)$
- (8) $\frac{n^2(n+1)^2}{4}$

2.12

- (1) $O(1)$
- (2) $O(1)$
- (3) $O(n)$
- (4) $O(n^2)$
- (5) $O(n)$
- (6) $O(n^2)$
- (7) $O(n^3)$
- (8) $O(n^4)$

1.13

- (1) 1부터 n 까지의 합을 구하는 함수
- (2) `sum=sum+1`

- (3) n 번
- (4) $O(n)$
- (5) 수열의 합 공식 이용: $O(1)$

1.14

- (1) 리스트에서 최댓값과 최솟값의 차이를 구하는 함수
- (2) if문
- (3) 리스트의 크기 n
- (4) $O(n)$
- (5) 없음

1.15

- (1) $O(n^2)$
- (2) $O(\log_2 n)$
- (3) $O(n^2)$
- (4) $O(\log_2 n)$

<2.4절>

1.16

- (1) 순환호출의 인수 : 문제의 크기가 줄지 않음 --> 무한루프
- (2) 순환호출에서 종료 조건이 없음 --> 무한루프

1.17 5번

1.18

5
4
3
2
1
0
반환 값 = 15

1.19 15개

1.20

(1) $3(n-1)$

(2) 2^{n+1}

(3) $n(n+1)/2$

(4) $\log_2 n$

(5) $\sum_{i=0}^k 3^i = \frac{3}{2}n - \frac{1}{2}$

1.21

(1) $F(n) = F(n-1) + 2n - 1, F(1) = 1, n^2$ 을 구하는 알고리즘

(2) $T(n) = T(n-1) + 1, T(1) = 0, O(n)$

(3) $T(n) = T(n-1) + 2, T(1) = 0, O(n)$