

1.1 다음 중 입력이 반드시 필요하다고는 볼 수 없는 알고리즘은?

- ① 배열에서 최댓값을 찾는 알고리즘
- ② 두 수의 최대 공약수를 구하는 알고리즘
- ③ 평면상에서 두 점 사이의 거리를 구하는 알고리즘
- ④ 5개의 로또 번호 예측하는 알고리즘 ?

1.2 다음 중 추상 자료형의 설명으로 옳지 않은 것은?

- ① 사용자들은 추상 자료형이 제공하는 연산만을 사용할 수 있다.
- ② 사용자들은 추상 자료형을 어떻게 사용하는지를 알아야 한다.
- ③ 사용자들은 추상 자료형 내부의 데이터를 직접 접근할 수 없다.
- ④ 사용자들은 추상 자료형 어떻게 구현되었는지 정확히 알아야 이용할 수 있다.  
↑ 인터페이스 제공

1.3 다음 중 파이썬에서 실행시간 측정을 위해 사용할 수 있는 모듈은?

- ① time
- ② date
- ③ numpy
- ④ copy

1.4 알고리즘 시간 복잡도  $O(1)$ 이 의미하는 것은?

- ① 컴퓨터 처리가 불가
- ② 입력 데이터 수가 한 개
- ③ 수행시간이 입력 데이터 수와 관계없이 일정
- ④ 알고리즘 길이가 입력 데이터보다 작음

1.5 다음 설명이 옳으면 ○, 틀리면 ×를 표시하라.

- ① 어떤 알고리즘의 효율성은 사용하는 자료구조와 밀접한 관련이 있다. ○
- ② 추상 자료형은 어떤 자료들과 자료에 가해지는 연산들을 구체적으로 표시하는데, 어떤(what?) 자료나 연산이 제공되는가 뿐만 아니라 이들이 어떻게(how?) 구현되는가도 정의한다. ×
- ③ 크기가  $n$ 인 배열의 모든 항목의 합을 구하는 알고리즘은 최선과 최악의 경우에 대한 시간 복잡도가 다르다. ○

- ④ 배열에 같은 항목이 있는지는 판단하는 알고리즘은 최선과 최악의 경우에 대한 시간 복잡도가 다르다. ○
- ⑤ 배열에 같은 항목이 있는지는 최악의 경우에도  $O(n)$ 에 판단할 수 있다. ✕

1.6 다음의 시간 복잡도 함수를 빅오 표기법으로 나타내라.

(1)  $T(n) = n^2 + 10n + 8$

(2)  $T(n) = n^3 + 10000n^2 + 50n$

(3)  $T(n) = n^2 \log_2 n + n^3 + 3$

(4)  $T(n) = 7(2^n) + 3^n$

(5)  $T(n) = 3^n + n!$

1.7 다음의 빅오 표기법들을 실행시간이 적게 걸리는 것부터 나열하라.

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

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

1.8 다음 알고리즘의 시간 복잡도를 빅오 표기법으로 나타내라.  $O(2n^2) < O(n!)$

def algorithm1(n) :

def algorithm2(n) :

1.10 다음과 같은 순환적인 프로그램에서 sub(3)과 같이 호출할 때 함수 sub()가 호출되는 횟수는?

5/94리 1:1  
2:1  
3:1  
2

```
def sub(n) :  
    if n <= 1 :  
        return n  
    return sub(n-1) + sub(n-2)
```

1.11 다음 함수에서 asterisk(5)와 같이 호출할 때 출력되는 \*의 개수는?

```
def asterisk(i) :  
    if i > 1 :  
        asterisk(i/2)  
        asterisk(i/2)  
    print("*", end="")
```

\* 1개 출력