**1. 다음 알고리즘들에 대해 입력의 크기와 기본 연산을 설명해 보라.**

(1) 입력 리스트에서 가장 큰 항목을 찾는 알고리즘

입력의 크기 = 입력 리스트의 크기 -> O(n)

기본 연산 = 비교 연산

(2) 입력 리스트의 모든 숫자를 더하는 알고리즘

입력의 크기 = 입력 리스트의 크기 -> O(n)

기본 연산 = 더하기 연산

(3) n! 알고리즘

입력의 크기 = n -> O(n)

기본 연산 = 곱하기 연산

(4) 두 개의 nXn 행렬을 더하는 알고리즘

입력의 크기 = n -> O(n)

기본 연산 = 더하기 연산

(5) 유클리드 알고리즘(알고리즘 1.7)

입력의 크기 = gcb(a, b) -> a, b

기본 연산 = 나머지 연산

(6) 최대 공약수 알고리즘(알고리즘 1.5)

입력의 크기 = a, b -> O(a\*b)

(7) 최대 공약수 알고리즘(알고리즘 1.6)

입력의 크기 = a, b -> O(a)

**2. 위 문제의 알고리즘들에 대해 입력의 구성에 따른 처리시간(최선, 평균, 최악) 특성을 각각 설명하라.**

(1) 최선, 평균, 최악으로 나눌 수 없음

(2) 최선, 평균, 최악으로 나눌 수 없음

(3) 최선, 평균, 최악으로 나눌 수 없음

(4) 최선, 평균, 최악으로 나눌 수 없음

(5) 처리시간의 특성을 계산하기 어려움

(6) 처리시간의 특성을 계산하기 어려움

(7) 처리시간의 특성을 계산하기 어려움

**3. 다음의 두 알고리즘이 입력의 크기에 따라 어떤 증가 속도를 갖는지를 설명하고, 입력의 구성에 따라 처리시간이 달라지는지를 설명하라.**

(1) 입력 리스트에서 어떤 항목이 있는지를 검사하는 알고리즘

최선의 경우 = O(1) -> 찾는 값이 맨 앞에 있는 경우

최악의 경우 = O(n) -> 찾는 값이 마지막에 있거나 없는 경우

(2) 입력 리스트에서 어떤 항목이 몇 번 나타나는지를 구하는 알고리즘

무조건 끝까지 검사를 하기에 최선, 평균, 최악의 구분이 없음

= O(n)

**4. 다음 시간 복잡도 함수들의 증가 속도를 비교하고, =, >, < 기호로 표시하라.**

(1) = (2) =

(3) < (4) =

(5) <

**5. 입력의 크기가 두 배로 늘어나면 다음 시간 복잡도 함수들의 처리시간은 몇 배가 늘어날까?**

(1) 1+log22n/log2n (2) 2 (3) 4 (4) 4 (5) 루트2

**6. 다음의 시간 복잡도 함수를 빅오 표기법으로 나타내라.**

(1) O(n^2) (2) O(n^3)

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

(5) O(n!)

**7. 다음의 빅오 표기법들을 실행 시간이 적게 걸리는 것부터 나열하라.**

O(1) – O(log n) – O(n) – O(n log n) – O(n^2) – O(n^3) – O(2^n) – O(n!)

**8 다음의 시간 복잡도 함수들을 증가 속도의 오름차순으로 정렬하라.**

log2 n – In^2\*n – 0.0001n^4+3n+1 – 3^n – 4^n – 2^3n

**10. 똑같이 생긴 n개의 동전과 양팔 저울이 있다. 동전 중에서 하나는 가짜 동전이고 진짜 동전과 무게가 약간 다르다. 양팔 저울은 동전의 무게를 직접 측정할 수 없고, 양쪽의 무게가 같은지 또는 어느 쪽이 더 무거운지를 알 수 있을 뿐이다. 이 저울을 이용해 O(1)에 위조 동전이 진짜 동전보다 무거운지 가벼운지를 판단하는 알고리즘을 작성하라. 단, n>2이다.**

n개의 동전을 반으로 나누고 홀수인 경우엔 동전 하나를 제외한다 – 양팔 저울로 나눈 동전의 무게를 측정한다 – 만약 무게가 같으면 제외한 동전과 그냥 동전 하나의 무게를 비교하여 가벼운지 무거운지를 알 수 있다 – 같지 않다면 무거운 그룹을 다시 반으로 나눠서 무게를 잰다 – 두 그룹이 같으면 가짜 동전은 가볍고 다르면 가짜 동전은 무겁다

**12. 문제 11의 (3)~(8)에 대한 시간 복잡도를 빅오 표기법으로 나타내라.**

(3) O(n) (4) O(n^2)

(5) O(n) (6) O(n^2)

(7) O(n^3) (8) O(n^4)

**13. 다음 알고리즘에 대해 물음에 답하라.**

(1) 1부터 n까지 더하기 위함

(2) sum = sum + i

(3) n번

(4) O(n)

(5) O(1)

**14. 다음 알고리즘에 대해 문제 13의 (1)~(5)에 답하라.**

(1) max 값과 min 값을 동시에 구하고 max-min 값을 반환하기 위함

(2) max < A[i], min > A[i]

(3) 2(n-1)

(4) O(n)

(5) 없음

**15. 다음 알고리즘의 시간 복잡도를 빅오 표기법으로 나타내라.**

(1) O(n^2) (2) O(log2^n)

(3) O(n^2) (4) O(log2 n)

**16. 다음의 순환 알고리즘에서 잘못된 점을 찾아라.**

(1) n이 줄어들지 않기 때문에 무한히 반복된다

(2) if 종료 조건문이 없음

**17. 다음의 순환 알고리즘을 sub(3)과 같이 호출할 때 sub()가 호출되는 전체 횟수를 구하라.**

- 5번

**18. 다음 함수를 sum(5)로 호출하였을 때, 화면에 출력되는 내용과 함수의 반환 값을 구하라.**

- 15

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

- 15개

**20. 다음의 순환 관계식을 풀어라.**

(1) T(n) = 3(n-1)

(2) 2^n+1

(3) n(n+1)/2

(4) log2 n+1

(5) (3/2)n-1/2

**21. 다음과 같은 순환 알고리즘이 주어졌다. 물음에 답하라.**

(1) n^2을 구하는 알고리즘이다

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

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