자료구조 1장 연습문제

01. 2개의 정수를 서로 교환하는 알고리즘을 의사코드로 작성해보자.

```
change(A, B)
  tmp ← A
  A ← B
  B ← tmp
  return (A, B)
```

02. 사용자로부터 받은 2개의 정수 중에서 더 큰 수를 찾는 알고리즘을 의사코드로 작성해보자.

```
find_max (A, B)
    if A > B then
        return A;
    else
        return B;
```

03. 1부터 n까지의 합을 계산하는 알고리즘을 의사코드로 작성해보자.

```
get_sum(n)
  tmp ← 0;
  for i ← 1 to n do
      tmp ← tmp + i;
  return tmp;
```

04. Set 추상 자료형을 정의하라.

객체정의 : 집합은 원소라 불리우는 데이터 요소들의 모임

연산정의

Create() := 집합을 생성하여 반환한다.

Insert(S, item) := 원소 item을 집합 S에 저장한다.

Remove(S, item) := 원소 item을 집합 S에서 삭제한다.

Is_In(S, item) := 집합 S에 item이 있는지 검사한다.

Union(S_1, S_2) := S_1과 S_2의 합집합을 구한다.

Intersection(S_1, S_2) := S_1과 S_2의 교집합을 구한다.

Difference(S_1, S_2) := S_1과 S_2의 차집합을 구한다.

06. 다음과 같은 코드의 시간 복잡도는??

```
for(i=1; i<n; i*=2)
    printf("Hello");
```

$n=2^t$ 라고 가정하자. $(t \in \mathbb{Z}_{\geq 0})$

i가 1부터 2씩 곱해서 $n(2^t)$가 되는 순간 프로그램이 종료되므로

총, t번의 연산이 필요하다.

즉, $\log_2 n$ 번 연산이 필요하므로

$O(\log n)$ 의 복잡도를 갖는다.

07. 다음과 같은 코드의 시간복잡도는??

```
for(i=0; i<n; i++)
    for(j=1; j<n; j*=2)
        printf("Hello");
```

① 안쪽 for문

$n=2^t$ 라고 가정하면, 총 t번의 연산이 진행되므로

$\log_2 n$ 의 연산이 진행된다.

② 바깥쪽 for문

총 n번 연산을 진행한다.

⇒ ①, ②에 의해

총 n번, $\log_2 n$ 의 연산을 진행하므로

$O(n \cdot \log n)$ 의 시간복잡도를 가진다.

08. 시간복잡도 함수 $n^3+10n+8$을 빅오표기법으로 나타내면??

$n^3+10n+8 < 19 \cdot n^3 \quad (n>1)$

$\Rightarrow n^3+10n+8 \in O(n^3)$

∴ (3)

09. 시간복잡도 함수가 $7n+10$ 이라면 이것이 나타내는 것은 무엇인가??

시간복잡도는 입력데이터의 크기가 n일 때, 알고리즘을 이루고 있는 연산이 몇 번 수행되는지를 의미한다. ∴ (1)

10. $O(n^2)$의 시간복잡도를 가지는 알고리즘에서 입력의 개수가 2배로 되었다면, 실행 시간은 어떤 추세로 증가하는가??

$n=t$ 일 때, t^2의 연산이 필요하다

$n=2t$ 일 때, $4t^2$의 연산이 필요하다

∴ $O(n^2)$

⇒ 4배로 증가. ∴ (3)

11. $f(n)$에 대하여 엄격한 상한을 제공하는 표기법은 무엇인가?

빅오표기법. $\therefore (2)$

12. 다음의 빅오표기법들을 수행시간이 적게 걸리는 것부터 나열하라.

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

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

$O(1),\ O(\log n),\ O(n),\ O(n\log n),$

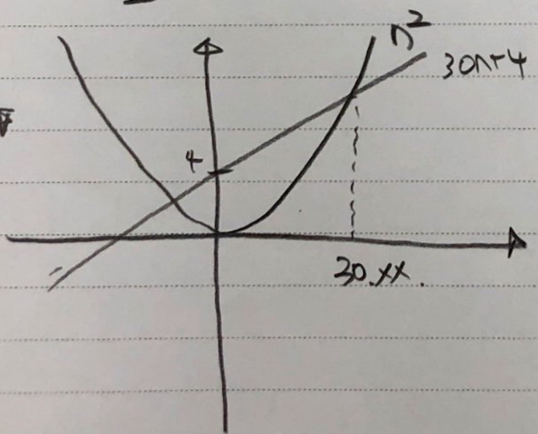
$O(n^2),\ O(2^n),\ O(n!)$

13. 두 함수 $30n+4$, n^2을 여러가지 n값으로 비교하라.
언제 $30n+4$가 n^2보다 작은 값을 갖는지를 구하라.
그래프를 그려보라.

$n^2 = 30n+4 \Rightarrow n^2 - 30n - 4 = 0$

$\Rightarrow \dfrac{30 \pm \sqrt{900+16}}{2} = 15 \pm \sqrt{229}$

$\Rightarrow$



n이 30보다 커지면 n^2이 더 커지고,
30보다 작거나 같으면 $30n+4$가 더 크다.

14. 다음은 실제로 프로그램의 수행시간을 측정하여 도표로 나타내는 시간복잡도를 예측하여 빅오 표기법으로 나타내라.

입력의 개수 n	수행시간(초)
2	2
4	8
8	25
16	63
32	162

$\Rightarrow$

입력의 개수 n	수행시간(초)
2	2×1
4	4×2
8	$\doteqdot 8\times 3$
16	$\doteqdot 16\times 4$
32	$\doteqdot 32\times 5$

$\Rightarrow$ 수행시간은 $n\times \log_2 n$ 과 유사하다.

$\therefore O(n\log n)$

15. 빅오표기법의 정의를 사용하여 다음을 증명하라.

$5n^2+3 = O(n^2)$

$5n^2+3 < 9n^2,\quad n > 1$

따라서,

$5n^2+3 \in O(n^2)$

16. 빅오표기법의 정의를 이용하여 $6n^2+3n$이 $O(n)$이 될 수 없음을 보여라.

$6n^2+3n \in O(n)$ 이라고 하자.

정의에 의해,

$\forall n \geq t$ 에 대해 $6n^2+3n \leq k \cdot n$ 인

k와 t가 존재한다.

$\Rightarrow 6n^2+3n-k \cdot n \leq 0$

$\Leftrightarrow 0 \leq n \leq \frac{k-3}{6}$

or

$\frac{k-3}{6} \leq n \leq 0$

하지만, $\forall n \geq t$ 에서 성립해야 하므로

모순이 생긴다.

($\because$ n의 상한이 생기므로)

$\Rightarrow 6n^2+3n \notin O(n)$

17. 배열에 정수가 들어있다고 가정하고, 다음의 작업의 최악, 최선의 시간복잡도를 표기하라.

(1) 배열의 n번째 숫자를 화면에 출력한다.

최선 : $O(1)$, 최악 : $O(1)$

(2) 배열안의 숫자중 최소값을 찾는다.

최선 : $O(n)$, 최악 : $O(n)$

(3) 배열의 모든 숫자를 더한다.

최선 : $O(n)$, 최악 : $O(n)$