- 코드를 분석해보면 수행되는 수행되는 연산들의 횟수를 입력 크기의 함수로 만들 수 있다.

```
ArrayMax(A,n)

  tmp ← A[0];                          1번의 대입 연산
  for i←1 to n-1 do                    루프 제어 연산은 제외
       if tmp < A[i] then              n-1번의 비교 연산
              tmp ← A[i];              n-1번의 대입 연산(최대)
  return tmp;                          1번의 반환 연산

                                       총 연산수= 2n(최대)
```

빅오 표기법

- 차수가 가장 큰 항이 가장 영향을 크게 미치고 다른 항들은 상대적으로 무시될 수 있음
  - 예: T(n) = $n^2 + n + 1$
    - n=1일때 : T(n) = 1 + 1 + 1 = 3 (33.3%)
    - n=10일때 : T(n) = 100 + 10 + 1 = 111 (90%)
    - n=100일때 : T(n) = 10000 + 100 + 1 = 10101 (99%)
    - n=1,000일때 : T(n) = 1000000 + 1000 + 1 = 1001001 (99.9%)

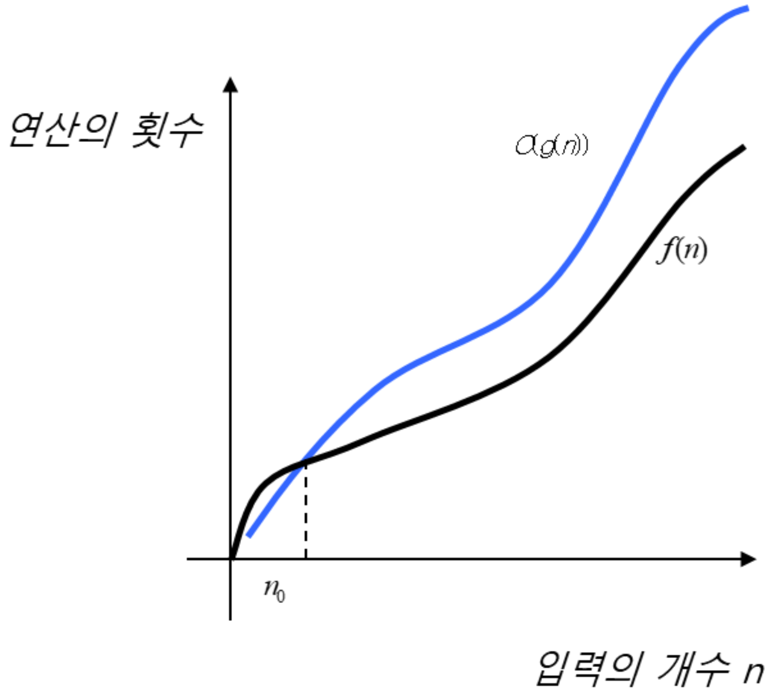
n=100인 경우

T(n)= n^2 + n + 1

99%

1%

- 두 개의 함수 f(n)과 g(n)이 주어졌을 때, 모든 n≥n0에 대하여 |f(n)| ≤ c|g(n)|을 만족하는 2개의 상수 c와 n0가 존재하면 f(n)=O(g(n))이다.
  - 연산의 횟수를 대략적(점근적)으로 표기한 것
  - 빅오는 함수의 상한을 표시한다.
  - (예) n≥5 이면 2n+1 <10n 이므로 2n+1 = O(n)



- 예제 1.1 빅오 표기법
  - $f(n)$=5이면 $O(1)$이다. 왜냐하면 $n0$=1, c=10일 때,
    n≥1에 대하여 5≤10·1이 되기 때문이다.

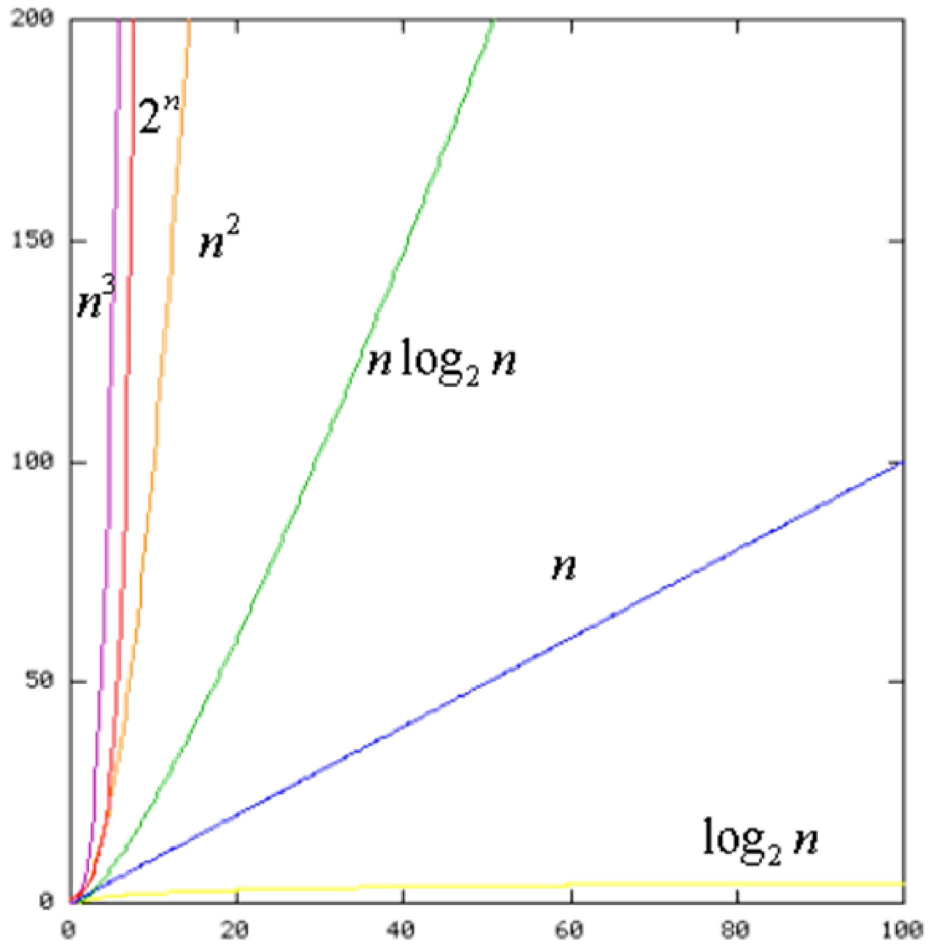
  - $f(n)$=2n+1이면 $O(n)$이다. 왜냐하면 $n0$=2, c=3일 때,
    $n \geq 2$에 대하여 2n+1≤ 3n이 되기 때문이다.

  - $f(n)=3n^2+100$ 이면 $O(n^2)$이다. 왜냐하면 $n0$=100, c=5일 때, $n \geq 100$에 대하여 $3n^2+100 \leq 5^2$이 되기 때문이다.

  - $f(n)=5 \cdot 2^n + 10n^2+100$ 이면 $O(2^n)$이다.
    왜냐하면 $n0$=1000, c=10일 때, $n \geq 1000$에 대하여 $5 \cdot 2^n + 10n^2+100 < 10 \cdot 2^n$이 되기 때문이다.

빅오 표기법의 종류

- O(1) : 상수형
- O(logn) : 로그형
- O(n) : 선형
- O(nlogn) : 로그선형
- O(n^2) : 2차형
- O(n^3) : 3차형
- O(n^k) : k차형
- O(2^n) : 지수형
- O(n!) : 팩토리얼형

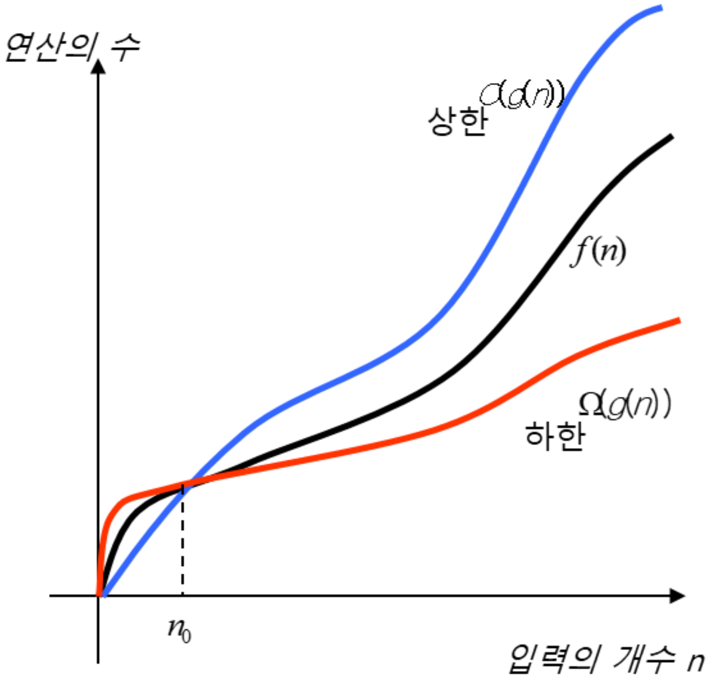


시간복잡도	n					
	1	2	4	8	16	32
1	1	1	1	1	1	1
$logn$	0	1	2	3	4	5
n	1	2	4	8	16	32
$nlogn$	0	2	8	24	64	160
n^2	1	4	16	64	256	1024
n^3	1	8	64	512	4096	32768
2^n	2	4	16	256	65536	4294967296
$n!$	1	2	24	40326	20922789888000	26313×10^{33}

빅오메가 표기법

- 정의
  - 모든 n≥n0에 대하여 |f(n)| ≥ c|g(n)|을 만족하는 2개의 상수 c와 n0가 존재하면 f(n)=Ω(g(n))이다.

  - 빅오메가는 함수의 하한을 표시
  - (예) n ≥ 1 이면 2n+1 > n 이므로 n = Ω(n)



빅세타 표기법

- 정의
  - 모든 n≥n0에 대하여 $c1|g(n)| \le |f(n)| \le c2|g(n)|$을 만족하는 3개의 상수 c1, c2와 n0가 존재하면 $f(n)=\theta(g(n))$이다.

  - 빅세타는 함수의 하한인 동시에 상한을 표시한다.
  - $f(n)=O(g(n))$이면서 $f(n)= \Omega(g(n))$이면 $f(n)=\theta(n)$이다.
  - (예) $n \ge 1$이면 $n \le 2n+1 \le 3n$이므로 $2n+1= \theta(n)$