

Big-O Notation

최태훈

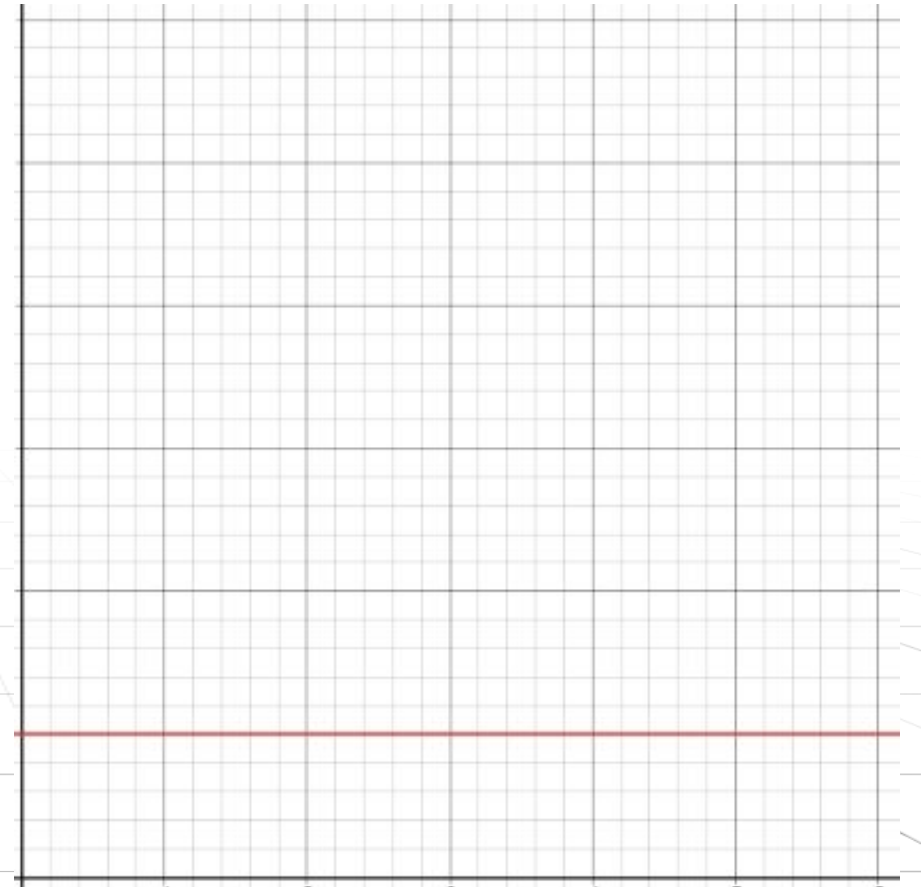
Big-O

- 알고리즘을 수학적으로 표현해주는 표기법
- 알고리즘의 시간과 공간 복잡도를 표현해주는 표기법
- 알고리즘의 실제 시간,공간을 표현해주는것 이 아닌 , 데이터의 증가율의 따른 알고리즘의 성능을 예측해주는 표기법

$O(1)$

- 입력데이터의 크기와 상관없이 항상 일정한 시간이 걸리는 알고리즘을 의미한다 .

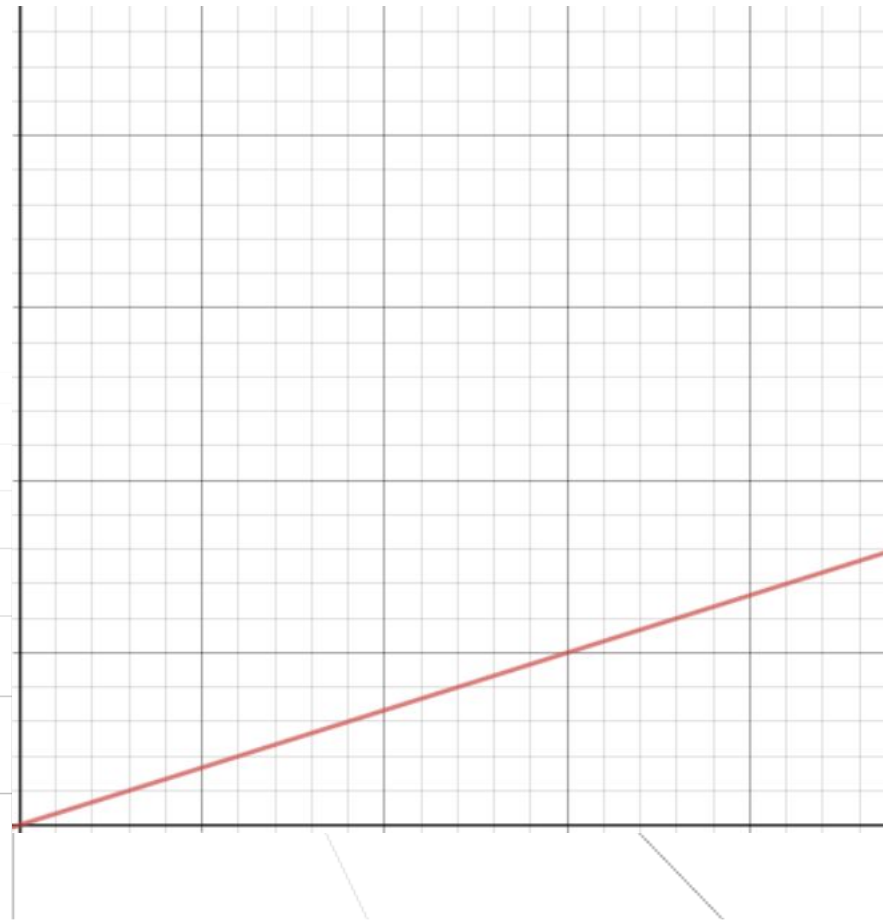
```
public class App {  
    public boolean firstnotation(int [] array) {  
        return (array[0]== 0) ? true : false ;  
    }  
}
```



$O(n)$

- 입력데이터의 크기에 비례해서 처리시간이 걸리는 알고리즘을 표현할때 사용
- Big-O** 표기법의 경우에는 문자앞에 붙은 계수를 신경쓰지않고 항상 계수를 제거한 문자만으로 표현한다

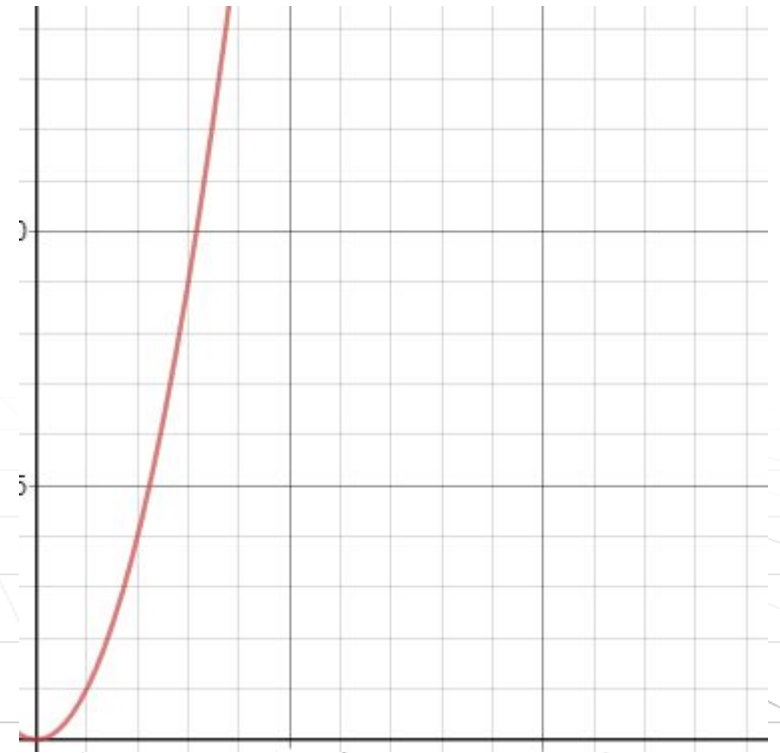
```
public void secondnotation(final int[] array) {  
    int sum = 0;  
    for (int i = 0; i < array.length; i++) {  
        sum += array[i];  
    }  
    int product = 0;  
    for (int i = 0; i < array.length; i++) {  
        product *= array[i];  
    }  
    System.out.println(sum + ", "+product);  
}
```



$O(n^2)$

-입력데이터의 크기의 제곱에 비례해서 처리시간이 걸리는 알고리즘을 표현할 때 사용

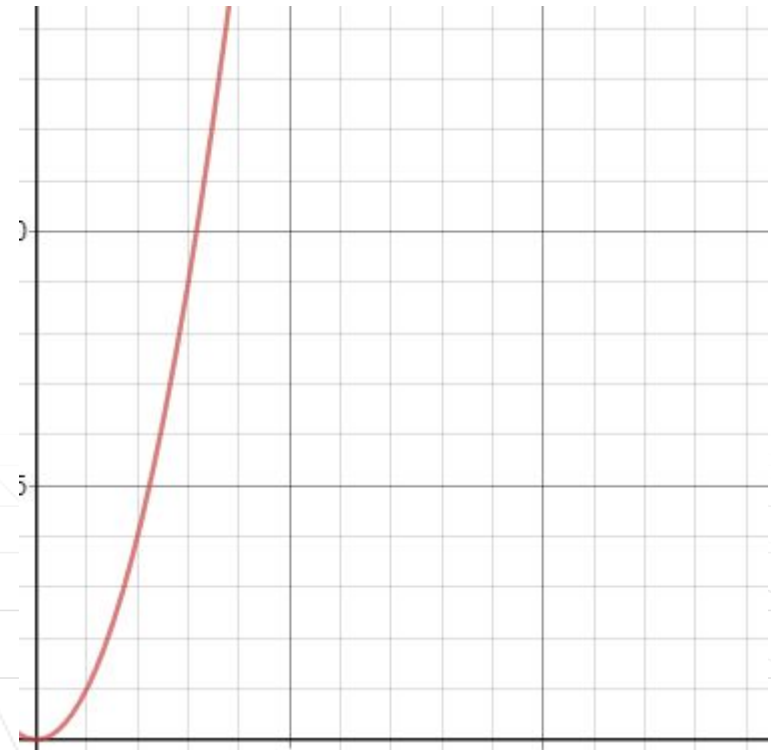
```
public void thirdnotation(final int[] array) {  
    for (int i = 0; i < array.length; i++) {  
        for(int j = 0; j<array.length; j++){  
            System.out.println(array[i]+"," array[i]);  
        }  
    }  
}
```



$O(mn)$

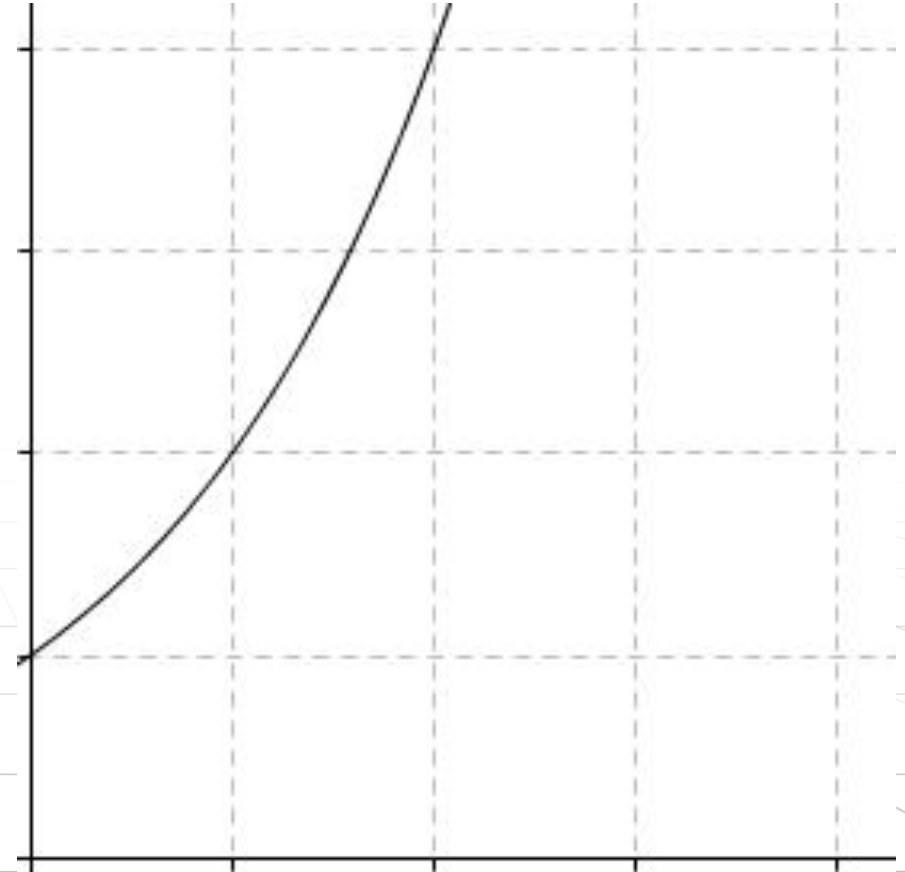
- “서로 다른” 두개의 입력이 주어지기때문에 $O(n^2)$ 이 아닌 $O(mn)$ 이다

```
public class App {  
    public void thirdnotation(int[] arrayA, int[] arrayB) {  
        for (int i = 0; i < arrayA.length; i++) {  
            for (int j = 0; j < arrayB.length; j++) {  
                System.out.println(arrayA[i] + "," + arrayB[j]);  
            }  
        }  
    }  
}
```



$O(2^n)$

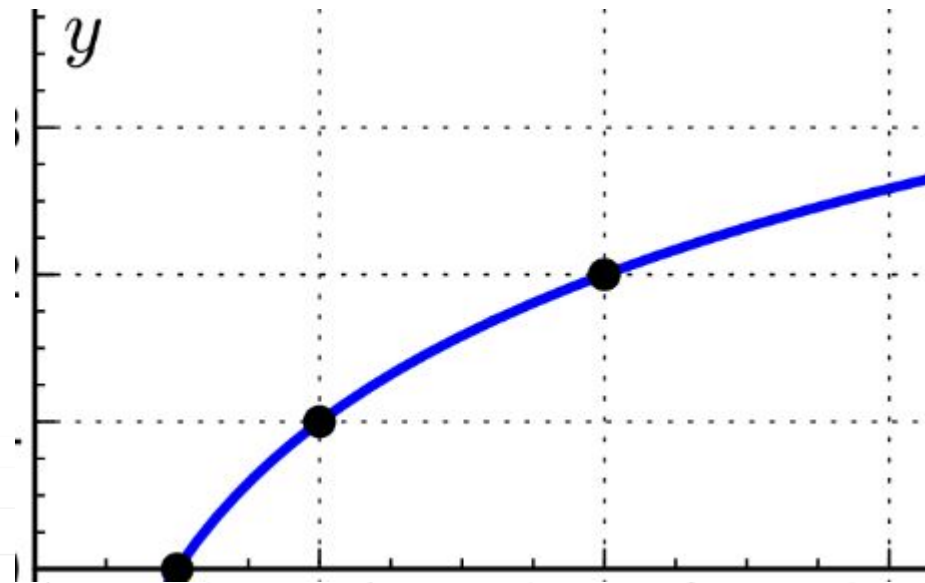
```
public int notation(int num) {  
    if(num <= 1)  
        return num;  
    else  
        return notation(num-1) + notation(num-1);  
}
```



$O(\log n)$

-Big-O 표기법의 경우에는 문자앞에 붙은 계수를 신경쓰지않고 항상 계수를 제거한 문자만으로 표현한다

```
public void notation(int key , int[] arr) {  
    int mid ;  
    int left = 0 ;  
    int right = arr.length -1;  
    while (right >= left){  
        mid = (right + left) / 2;  
        if( key == arr[mid]){  
            break;  
        }  
        if(key < arr[mid]){  
            right= mid -1;  
        }else {  
            left = mid + 1;  
        }  
    }  
}
```



중요한 사항

- $O(n^2+n)$ 과 같은 경우에는 $O(n^2)$ 으로 표기합니다 n 의 값이 커질수록 n^2 의 값이 n 의값 에 비하여 압도적으로 크기때문입니다 . 이럴 경우의 n^2 을 지배적인 항이라고 합니다 이렇게 여러개의 항이 같이있을때 지배적인 항이되는것은 그래프를 참고합니다

