

* 7/19 : (비트연산자, 비교연산자, 할당문, 메서드) 훈련
비트.이동연산자

comcs-java/comcs-java-lang/app



bitcamp-study/java-lang/app

com.comcs.quiz.ex01

Test01

Test02



복사-붙여넣기 연습할 것.

* 7/20 : (비트연산자, 비교연산자, 할당연산자, 메서드) 훈련
비트연산자

comcs-java/comcs-java-lang/app



bitcamp-study/java-lang/app

com.comcs.quiz.ex01

Test03

Test04

Test05



복사-붙여넣기 연습할 것.

* 비트 연산자의 특징

① OR

$$\begin{array}{r} 01 \\ 111 \\ \hline 11 \end{array} \leftarrow \text{부진}$$

$$\begin{array}{r} 01 \\ 100 \\ \hline 01 \end{array} \leftarrow \text{원래 값 통과}$$

② AND

$$\begin{array}{r} 01 \\ \& 11 \\ \hline 01 \end{array} \leftarrow \text{원래 값 통과}$$

$$\begin{array}{r} 01 \\ \& 00 \\ \hline 00 \end{array} \leftarrow \text{값 차단}$$

* value & 1
↳ 최하위 1비트 추출

$$\begin{array}{r} 01110011 \\ \& 00000001 \\ \hline 00000001 \end{array}$$

③ XOR

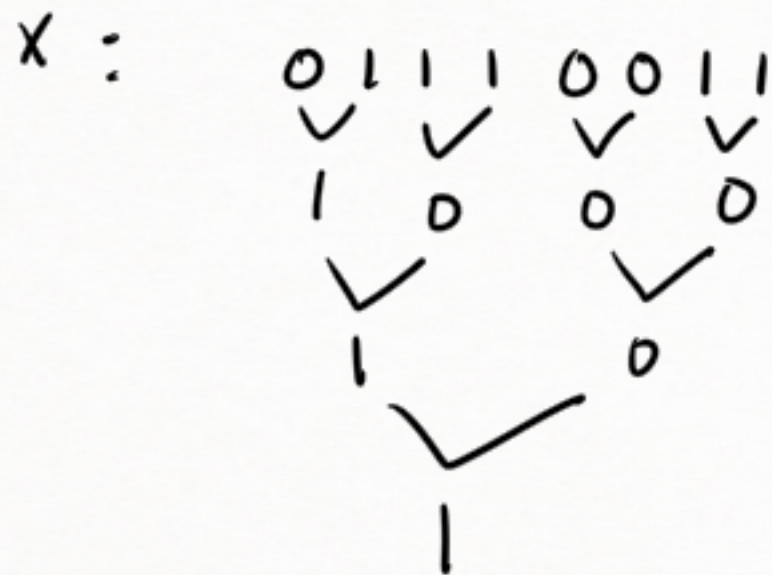
$$\begin{array}{r} 01 \\ 111 \\ \hline 10 \end{array} \leftarrow \text{원래 값을
스왑}$$

$$\begin{array}{r} 01 \\ 100 \\ \hline 01 \end{array} \leftarrow \text{원래 값을
그대로 통과}$$

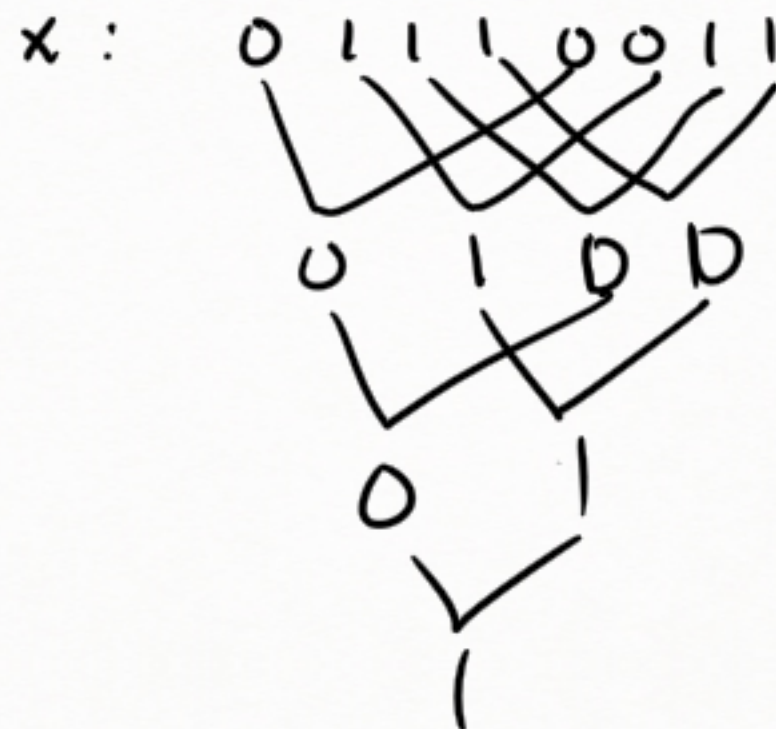
* msb et lsb

01110011
↑ msb most significant bit (최상위 비트)
↑ lsb least significant bit (최하위 비트)

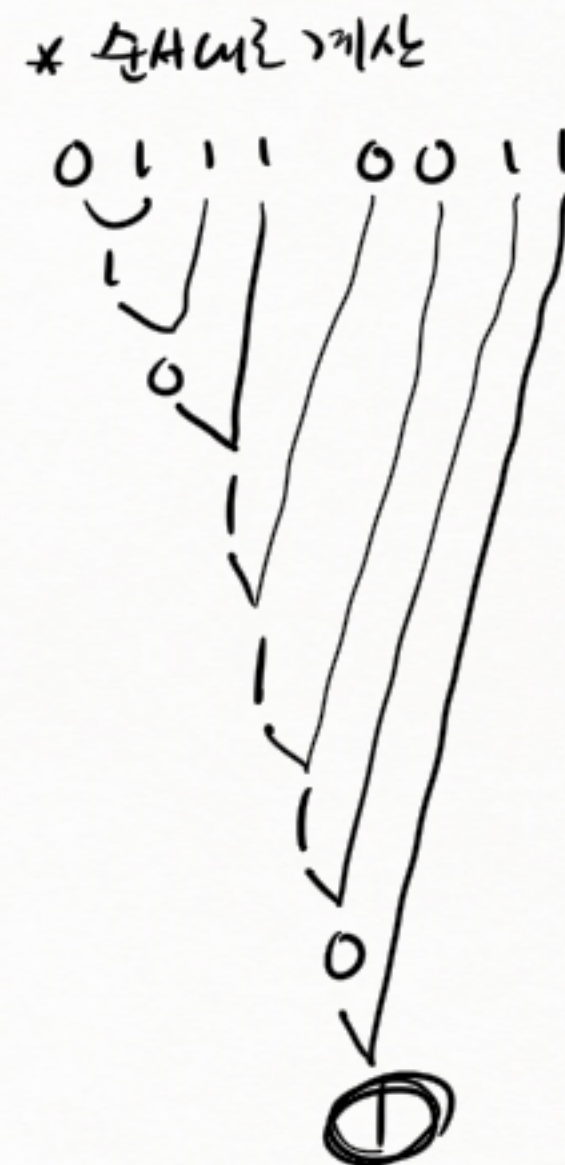
* XOR 과 비트 연산의 특징



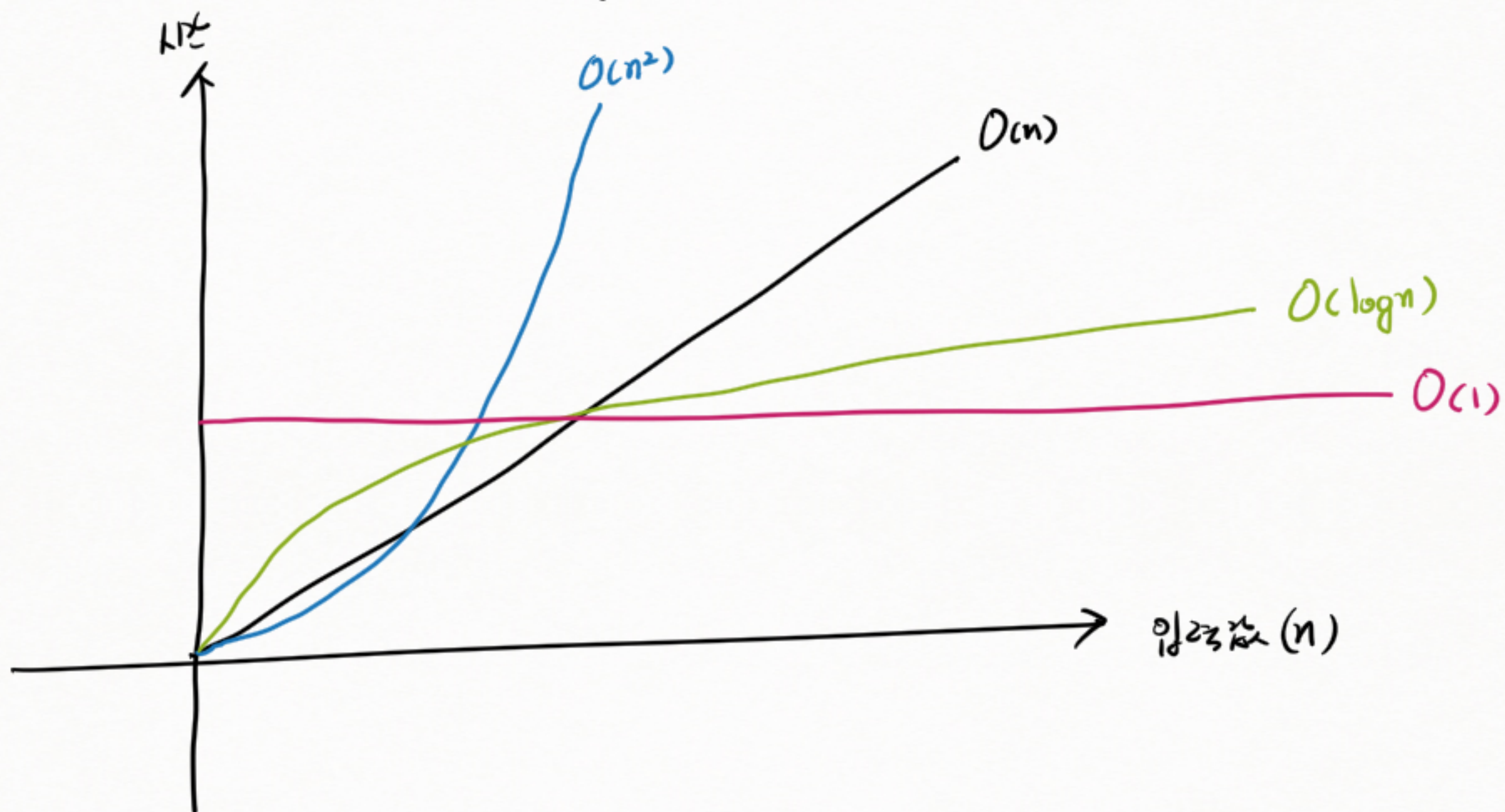
* 비트별 비교법을 반쪽



* 그룹별 비교법을 반쪽



* 시간 복잡도 : Big-O



* 패리티 비트 + XOR 연산

① $x \& (x-1)$

↳ 최하위 비트에서
1 값을 갖는 비트를
0으로 만든다

$$\begin{array}{r} 011001\textcolor{red}{\times}0 \quad x \\ \& 01100101 \quad (x-1) \\ \hline \end{array} \quad \textcircled{1}$$

$$\begin{array}{r} 01100\textcolor{red}{\times}00 \quad x \\ \& 01100011 \quad (x-1) \\ \hline \end{array} \quad \textcircled{2}$$

$$\begin{array}{r} 01\textcolor{red}{\times}00000 \quad x \\ \& 01011111 \quad (x-1) \\ \hline \end{array} \quad \textcircled{3}$$

$$\begin{array}{r} 0\textcolor{red}{\times}00000 \quad x \\ \& 00111111 \quad (x-1) \\ \hline 0\textcolor{red}{0000000} \end{array} \quad \textcircled{4}$$

$O(k)$

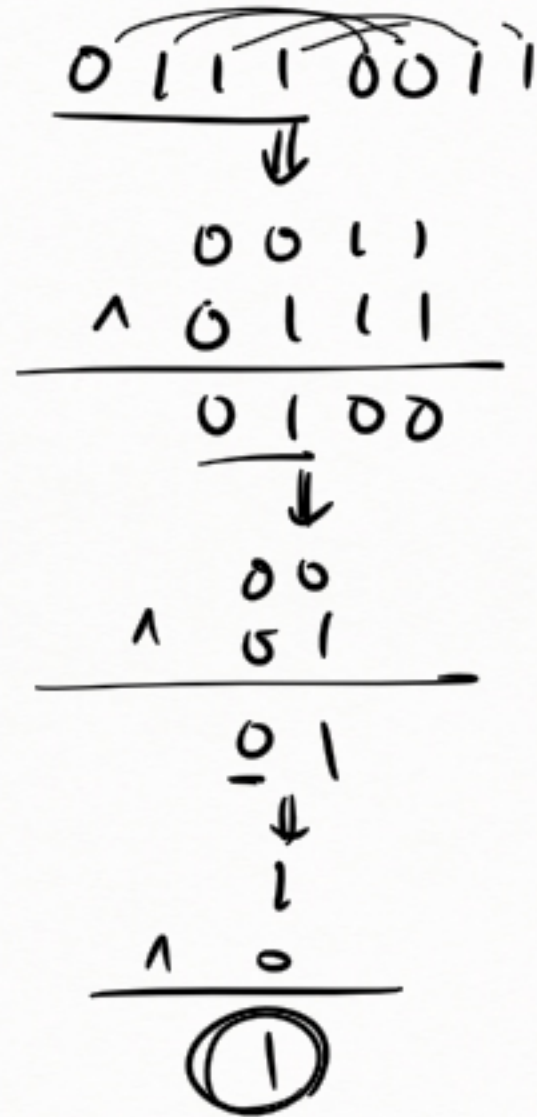
↳ k는 값이 1인 비트의 개수

* 두자리 비트의 XOR 연산

✓ 0과 1의 조합



✓ 3비트 XOR



0(1)

