

# 알고리즘 스터디 13회

---

비트마스킹이란

# CONTENTS

- 01 — 비트마스킹이란
- 02 — 비트마스킹을 하는 이유
- 03 — Java 기본 자료형
- 04 — 비트 연산자
- 05 — 비트마스킹 사용 예시

# 비트마스킹이란



## 비트마스킹(bitmasking)이란

컴퓨터가 데이터를 저장하는 방법은 0과 1로 비트를 저장하는 방법  
그 방법을 활용하여 사용하는 것이 비트마스킹

- ✓ 0, 1
- ✓ and, or, not, xor
- ✓ shift

# 02

## 비트마스킹을 하는 이유

머리가 복잡해지는 비트 마스킹 왜 사용할까요?

### 빠른 연산

덧셈, 뺄셈, 나눗셈 등은 기본적으로 비트 연산을 통해 이루어지므로 연산속도가 제일 빠름

### 메모리 최적화

32 크기를 가진 boolean배열과 int 하나랑 비교할 때 훨씬 메모리를 아낄 수 있음

### 빠른 집합 연산

and, or, not등을 활용하여 빠른 집합 연산이 가능합니다.

### 문제를 푸는 방식 중 하나

복잡하게 생각할 것 없이 알고리즘이라고 생각합시다.

# 03

## Java에서의 비트 수

<b>byte</b>	<b>1 바이트</b>	<b>-128 ~ 127</b>
<b>short</b>	<b>2 바이트</b>	<b>-32,768 ~ 32,767</b>
<b>int</b>	<b>4 바이트</b>	<b>-2,147,483,648 ~ 2,147,483,647</b>
<b>long</b>	<b>8 바이트</b>	<b>-9,223,372,036,854,775,808 ~ 9,223,372,036,854,775,807</b>

# 04

## 비트 연산자 - and

0	0	0
0	1	0
1	0	0
1	1	1

# 04

## 비트 연산자 - or

0	0	0
0	1	1
1	0	1
1	1	1

# 04

## 비트 연산자 - xor

0	0	0
0	1	1
1	0	1
1	1	0



# 04

## 비트 연산자 - not

0	1
1	0

# 04

## 비트 연산자 - left shift

1	1	2(10)
1	3	8(1000)
1	31	
1	32	

# 04

## 비트 연산자 - left shift

1	1	2(10)
1	3	8(1000)
1	31	-2147483648
1	32	

# 04

## 비트 연산자 - left shift

1	1	2(10)
1	3	8(1000)
1	31	-2147483648
1	32	1

# 04

## 비트 연산자 - right shift

2(10)	1	1
8(1000)	3	1
8(1000)	5	
-2147483648	30	
-2147483648	33	

# 04

## 비트 연산자 - right shift

2(10)	1	1
8(1000)	3	1
8(1000)	5	1
-2147483648	30	
-2147483648	33	

# 04

## 비트 연산자 - right shift

2(10)	1	1
8(1000)	3	1
8(1000)	5	1
-2147483648	30	-2
-2147483648	33	

# 04

## 비트 연산자 - right shift

2(10)	1	1
8(1000)	3	1
8(1000)	5	1
-2147483648	30	-2
-2147483648	33	-1



# 05

## 비트마스킹 사용 예시

### 비트를 통한 집합 구현

int를 사용하여 32개까지 저장 가능한 집합을 구현할 수 있습니다.

### 이분 탐색

이분 탐색을 할 때 right shift를 통해 중앙 값을 빠르게 구할 수 있습니다.

### DP

비트를 통해 구현한 집합으로 DP를 편하게 사용할 수 있습니다.

### 트리

부모와 자식 노드의 인덱스를 비트 연산을 통해 빠르게 구할 수 있습니다.