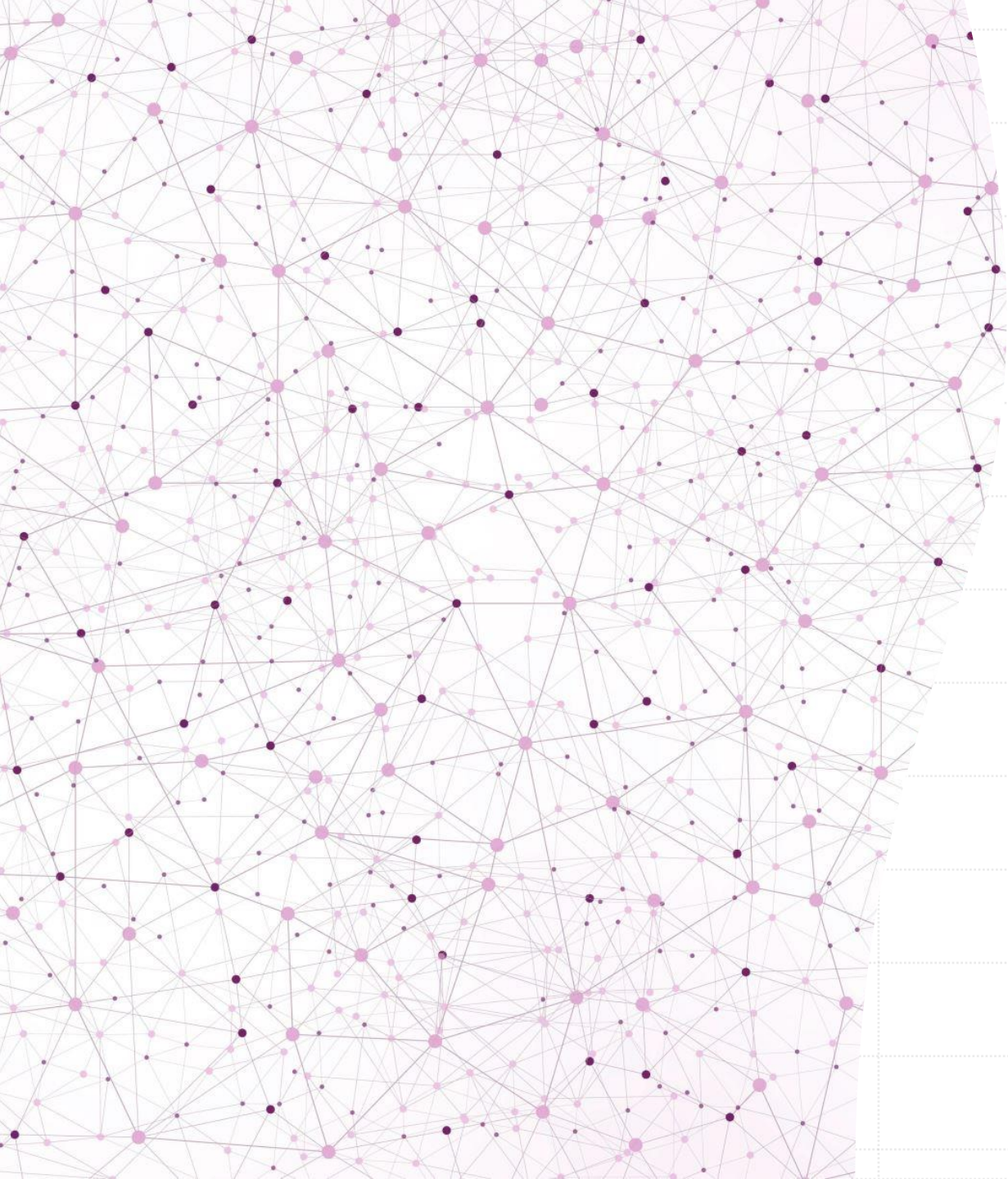


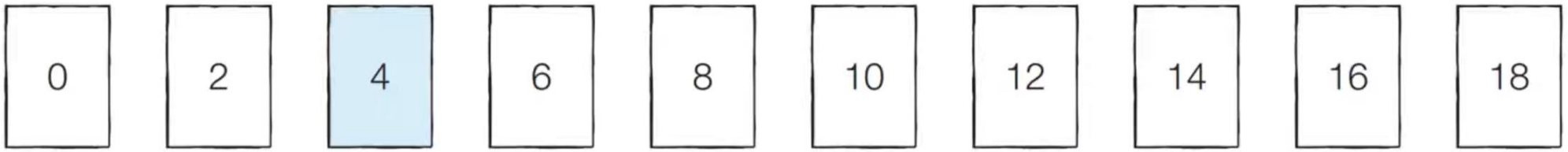
이진탐색

발표: 한혜원

이진 탐색

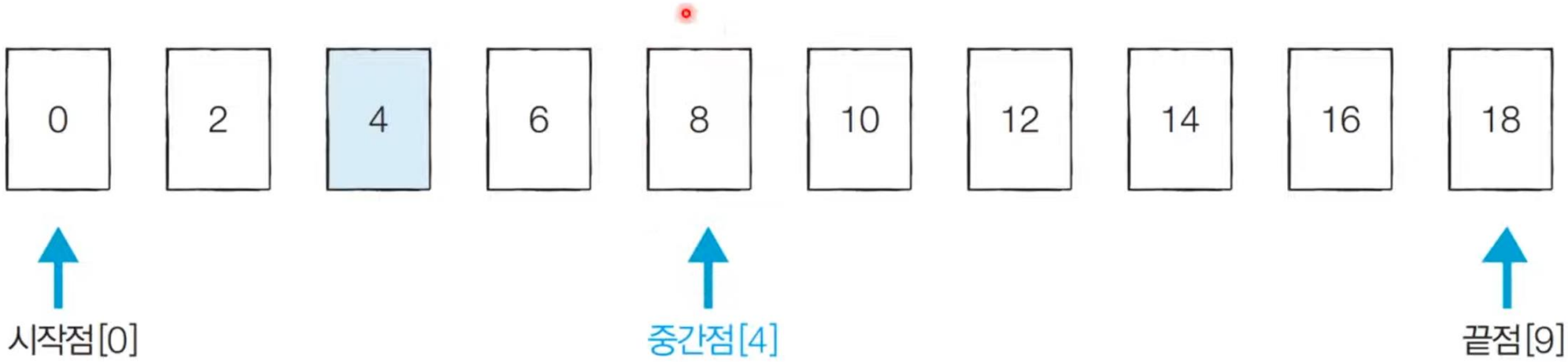
- 순차 탐색: 리스트 안에 있는 특정 데이터를 찾기 위해 앞에서부터 하나씩 확인
- 이진 탐색
  - 정렬되어 있는 리스트에서 탐색 범위를 절반씩 좁혀가며 확인
  - 시작점, 중간점, 끝점인 3개의 변수를 사용
  - O(log_2N)의 시간 복잡도를 갖음

- 이미 정렬된 10개의 데이터 중에서 값이 4인 원소를 찾는 예시를 살펴봅시다.

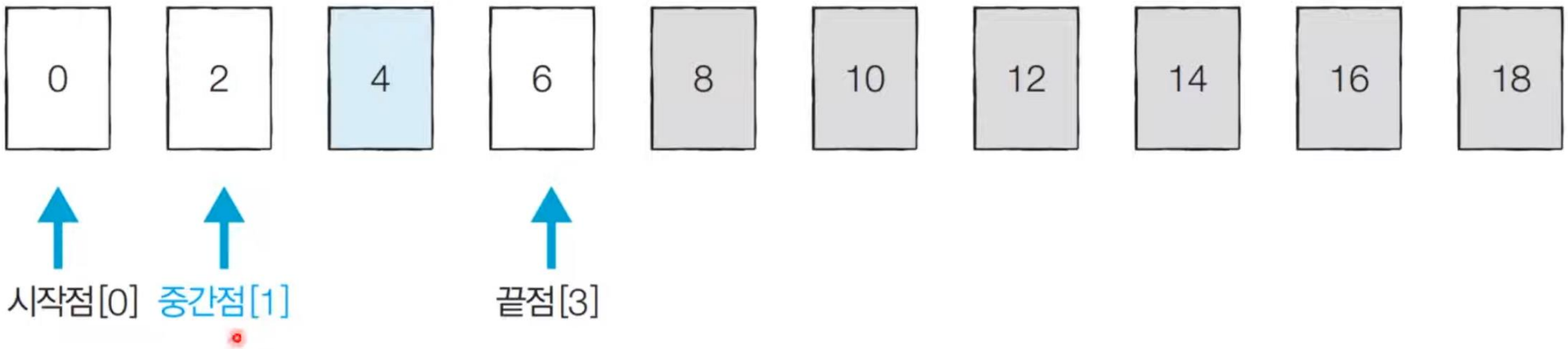


(출처: 나동빈 유튜브)

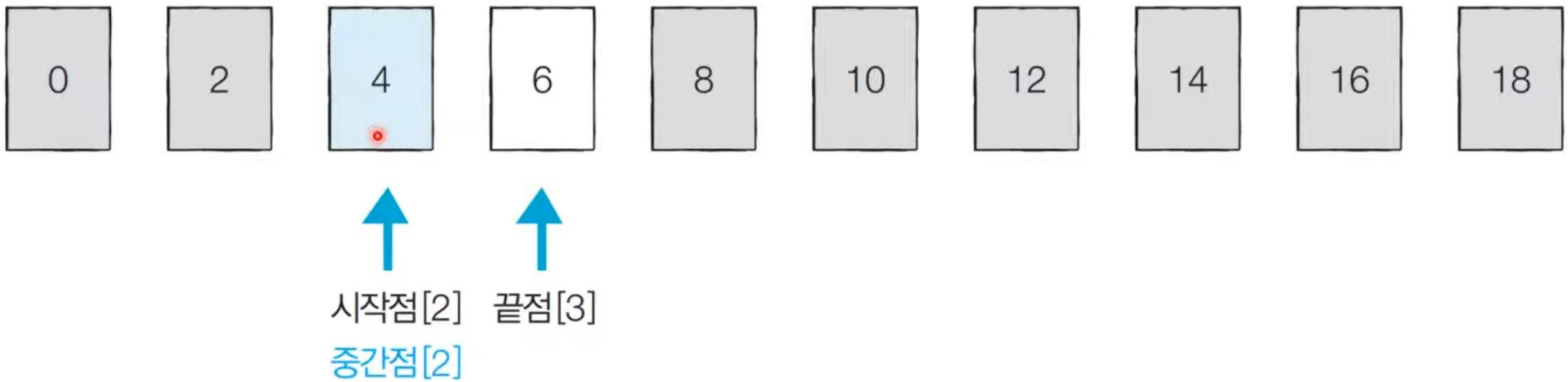
- **[Step 1]** 시작점: 0, 끝점: 9, 중간점: 4 (소수점 이하 제거)



- **[Step 2]** 시작점: 0, 끝점: 3, 중간점: 1 (소수점 이하 제거)



- **[Step 3]** 시작점: 2, 끝점: 3, 중간점: 2 (소수점 이하 제거)



파라메트릭 서치(parametric search)

- 최적화 문제를 결정 문제(예 혹은 아니오)로 바꾸어 해결하는 기법
- 코테에서 이진 탐색으로 풀어낼 수 있음

〈문제〉 떡볶이 떡 만들기: 문제 설명

- 오늘 동빈이는 여행 가신 부모님을 대신해서 떡집 일을 하기로 했습니다. 오늘은 떡볶이 떡을 만드는 날입니다. 동빈이네 떡볶이 떡은 재밌게도 떡볶이 떡의 길이가 일정하지 않습니다. 대신에 한 봉지 안에 들어가는 떡의 총 길이는 절단기로 잘라서 맞춰줍니다.
- 절단기에 **높이(H)**를 지정하면 줄지어진 떡을 한 번에 절단합니다. 높이가 H보다 긴 떡은 H 위의 부분이 잘릴 것이고, 낮은 떡은 잘리지 않습니다.
- 예를 들어 높이가 19, 14, 10, 17cm인 떡이 나란히 있고 절단기 높이를 15cm로 지정하면 자른 뒤 떡의 높이는 15, 14, 10, 15cm가 될 것입니다. 잘린 떡의 길이는 차례대로 4, 0, 0, 2cm입니다. 손님은 6cm 만큼의 길이를 가져갑니다.
- 손님이 왔을 때 요청한 총 길이가 M일 때 **적어도 M만큼의 떡을 얻기 위해 절단기에 설정할 수 있는 높이의 최댓값을 구하는 프로그램**을 작성하세요.

입력 예시

```
4 6
19 15 10 17
```

4 6 떡의 개수 N / 요청한 떡의 길이 M

19 15 10 17 떡의 개별 높이

출력 예시

```
15
```

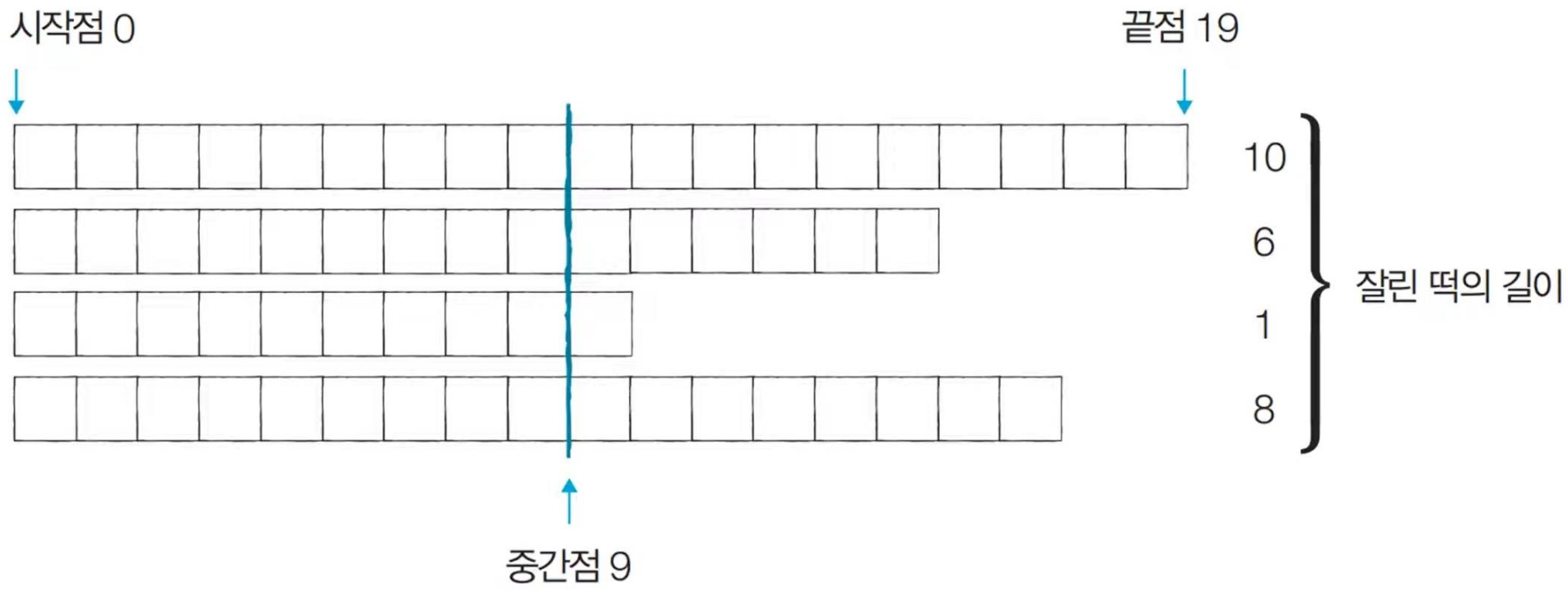
15 얻고 싶은 건 H의 최댓값

해결 아이디어

- 현재 이 높이(H)로 자르면 조건을 만족할 수 있는가?를 확인한 뒤 '예', '아니오'에 따라 탐색 범위를 좁혀서 해결 가능
- 보통 0부터 10억까지의 정수 같이 큰 탐색 범위를 보면 이진 탐색을 떠올려야 함

〈문제〉 떡볶이 떡 만들기: 문제 해결 아이디어

- **[Step 1]** 시작점: 0, 끝점: 19, 중간점: 9 이때 필요한 떡의 크기: M = 6이므로, **결과 저장**



이진 탐색 문제

- https://programmers.co.kr/learn/courses/30/lessons/43238

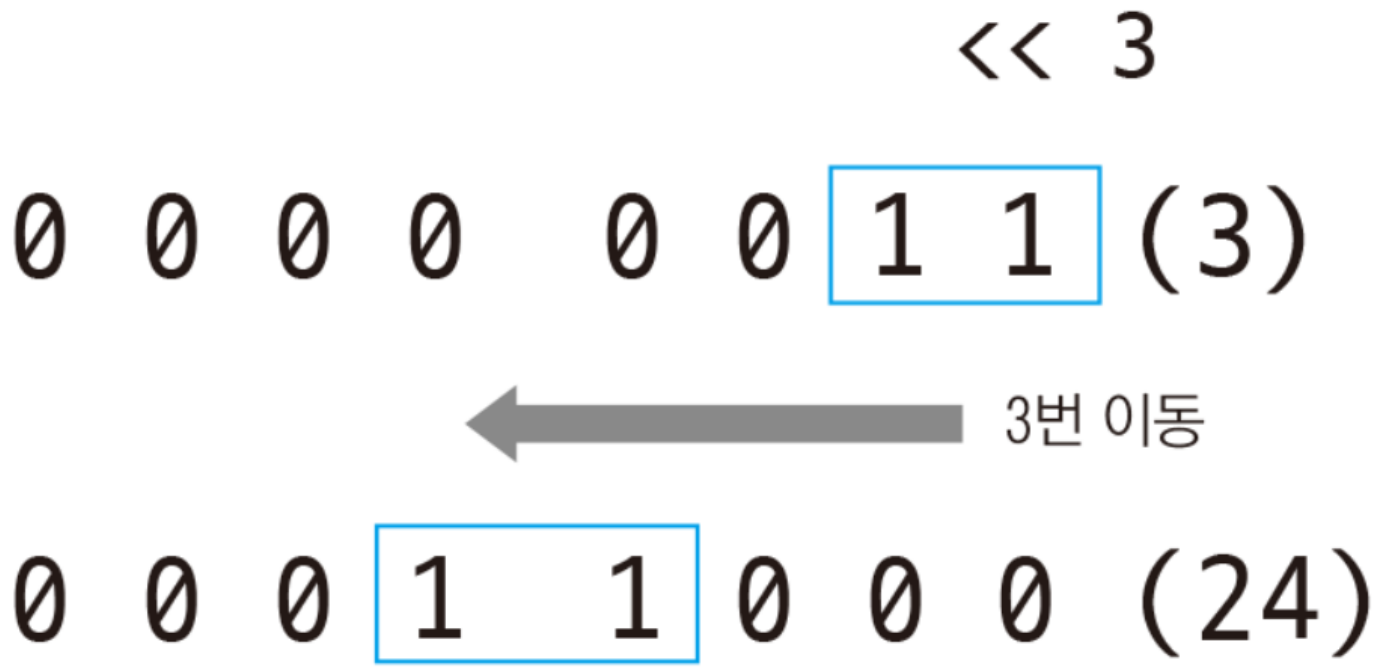
비트 조작

```
0000 0001(1)
0000 0011(3)
           ^
___________
0000 0010(2)
```

- 비트 연산자를 사용해야하는 문제도 있음
- 대표적 연산자로 & | ^ ~ >> <<
  - a&b 는 a와 b를 각각 2진수로 표현한 후 자릿수 별로 and 연산 (모두 1일 때 1)
  - a|b는 마찬가지로 자릿수 별로 or연산 (하나만 1 이라도 1)
  - a^b는 마찬가지로 자릿수 별로 두 비트가 다르면 1, 같으면 0으로 연산
  - ~a는 2진수로 표현한 후 각 자리의 비트를 0은 1로, 1은 0으로 반전

쉬프트 연산자 >>, <<

- 변수의 비트를 지정한 횟수만큼 이동함
- 3<<3의 예시 : 3을 비트로 표현하면 011, 왼쪽으로 비트를 3번 이동, 빈공간은 0으로 채움



비트 마스킹

- 집합의 배열의 인덱스를 비트로 표현할 수 있음
- 원소수가 적을 때 작은 메모리와 빠른 수행시간으로 해결 가능

배열 [1, 2, 3, 4]가 있을 때 이 배열의 부분집합을 구하면 다음과 같다
[],[1],[2],[3],[4],[1,2],[1,3] … [1,2,3,4]
이 경우, 비트마스킹을 해서 배열의 각 요소를 인덱스로 표현하여 접근할 수 있다

[1,2,3,4] -> 1111
[1,3,4] -> 1011
[1,2] -> 1100
[4] -> 0001
[] -> 0000

이때 1111을 10진수로 변환하면 15이다
즉, 0 ~ 15로 부분집합을 표현할 수 있고(정수로 표현 가능), 이 때 부분집합의 갯수는 16개이다

비트 연산 이용한 문제 원한다면

- https://programmers.co.kr/learn/courses/30/lessons/17681

- 풀이 https://onlydev.tistory.com/57