



최빈값 (가장 자주 나오는 수)

Input 배열, 빈도

ex) { 1, 1, 2, 2, 3, 3, 3, 5, 7, 8 }

인덱스 위치와 값이 같다면. (이름은 array 라고 하자)

새로운 배열을 하나 만들면서, 숫자의 범위에 따라
사이드를 정해줄 것임.

예를 들어 인덱스 배열의 값이 0 ~ 1000 까지라고 하면
1001 개 짜리 배열을 만들어 주면 됨. \Rightarrow 이름을 newarr 이라 하자

왜 그러냐 \Rightarrow 아래에서 설명함.

인덱스 배열은 선회하면서 해당 숫자를, 현재가 새로 만든 1001 개 짜리 배열의

인덱스로 접근함. (array.length())

for (int i = 0; i < 인덱스 배열 사이즈; i++) {
(새로 만들 배열) newarr[array[i]]++; }

이렇게 해주면 됨. 그러면 결국

newarr index

new arr	0
11	1
11	2
11	3
11	4
11	5
11	6
11	7
11	8

✓ ✓ 2개
 ✓ ✓ 2개
 ✓ ✓ ✓ 3개 3개 ✓ 최빈값.

✓ 1개

✓ 1개

✓ 1개

이런 식이 될 것이다.

* 단점
 (배열이 크면 클수록 메모리 사용량
 배열 인덱스에 큰 소모가.

* 근데 직관적이지, 이해하기 힘들.

결국 가장 큰 값을 가진 인덱스의 번호가
 가장 자주 나오는 수가 될 것이다.