문제

정렬된 두 묶음의 숫자 카드가 있다고 하자. 각 묶음의 카드의 수를 A, B라 하면 보통 두 묶음을 합쳐서 하나로 만드는 데에는 A+B 번의 비교를 해야 한다. 이를테면, 20장의 숫자 카드 묶음과 30장의 숫자 카드 묶음을 합치려면 50번의 비교가 필요하다.

매우 많은 숫자 카드 묶음이 책상 위에 놓여 있다. 이들을 두 묶음씩 골라 서로 합쳐나간다면, 고르는 순서에 따라서 비교 횟수가 매우 달라진다. 예를 들어 10장, 20장, 40장의 묶음이 있다면 10장과 20장을 합친 뒤, 합친 30장 묶음과 40장을 합친다면 (10 + 20) + (30 + 40) = 100번의 비교가 필요하다. 그러나 10장과 40장을 합친 뒤, 합친 50장 묶음과 20장을 합친다면 (10 + 40) + (50 + 20) = 120 번의 비교가 필요하므로 덜 효율적인 방법이다.

N개의 숫자 카드 묶음의 각각의 크기가 주어질 때, 최소한 몇 번의 비교가 필요한지를 구하는 프로그램을 작성하시오.

요구분석

최소한 몇번의 비교가 필요한지를 구하려고 한다.

카드 묶음을 합치고 다시 더하는 것을 반복해야 한다.

최소 값부터 더해 나가면 가장 적은 수만큼 비교할 수 있을 것이다.

최소힙을 이용하겠다.

이를 이용하면 최소값을 적은 시간 복잡도로 꺼낼 수 있다.

1. 값 입력하기

먼저 카드 묶음의 개수 N과 카드 묶음 각각의 크기를 입력받는다.

BufferedReader 로 입력받는다.(시간 복잡도 고려)

var reader = new BufferedReader(new InputStreamReader(System.in)); var tokenizer = new StringTokenizer(reader.readLine()); int N = Integer.parseInt(tokenizer.nextToken()); int M = Integer.parseInt(tokenizer.nextToken());

2. 최소힙 만들기

최소힙을 만들기 위해 PriorityQueue class를 이용한다.

값을 삽입하겠다.

3. 최소힙 최소값 제거하며 더하기 연산

4. 결과값 출력