

백준 2343 기타레슨(1)

2025-05-25 ~ 17:00

compare결과를 이용해서 최소 블루레이 용량을 찾아내야 함
파라메트릭 서치 탐색 함수 binarySearch

가능한 블루레이 용량의 범위에서, compare(mid) 결과를 이용해,
M개 이하의 블루레이로 가능한 가장 작은 용량을 이진 탐색으로 찾기

탐색 범위 설정

Int left = 가장 긴 강의 길이

Int right = 모든 강의 길이의 합

Left는 강의 중 가장 긴 것 -> 그보다 작으면 절대 불가능

Right는 모든 강의를 하나에 담는 경우 -> 가능한 최대치

이진 탐색 함수

Left와 right가 만나거나 교차할 때까지 반복

현재 시도해보는 블루레이 용량 $mid = (left + right) / 2$

이 mid로 나눴을 때 필요한 블루레이 수 계산 $result = compare(mid)$

compare결과에 따라 이진 탐색의 방향을 정함

result가 0보다 작거나 같으면 정답후보 or 더 줄이고

else, 더 크게 mid + 1

```
while (left <= right) {
    int mid = (left + right) / 2;
    int result = compare(mid);

    if (result <= 0) {
        // 블루레이 개수가 M 이하 → 정답 후보, 더 줄일 수 있음
        answer = mid;
        right = mid - 1;
    } else {
        // 블루레이 개수가 M 초과 → 용량이 작음 → 더 크게
        left = mid + 1;
    }
}
```

Main

int[] A 강의 길이를 저장할 배열

int M 사용할 블루레이 수 (입력)

➔ A, M static으로 compare와 binarySearch에도 사용

입력 받고 N, M = sc.nextInt() 강의 개수, 블루레이 개수

강의 길이 저장할 배열 생성 A = new int[N];

강의들을 하나씩 입력 받고,

가장 긴 강의 left로 (최소한 이 정도는 들어가야 가능), 총 길이 합 right로 (모두 한장에 넣는 최대의 경우)

```
package homework11_1;

import java.util.*;

public class Main {

    static int[] A;
    static int M;

    public static void main(String[] args) {
        try (Scanner sc = new Scanner(System.in)) {

            // 입력 받기
            int N = sc.nextInt(); // 강의 수
            M = sc.nextInt(); // 블루레이 개수
            A = new int[N];

            int left = 0, right = 0;
            for (int i = 0; i < N; i++) {
                A[i] = sc.nextInt();
                left = Math.max(left, A[i]); // 가장 긴 강의가 최소 용량
                right += A[i]; // 모든 강의 합이 최대 용량
            }

            int answer = binarySearch(left, right);
            System.out.println(answer);
        }
    }

    // 이진 탐색으로 최소 블루레이 용량 구하기
```

```

static int binarySearch(int left, int right) {
    int result = right; // 최소값 찾기이므로 최대치로 초기화

    while (left <= right) {
        int mid = (left + right) / 2;
        int cmp = compare(mid);

        if (cmp <= 0) {
            // M개 이하로 나눌 수 있음 → 줄여볼 수 있음
            result = mid;
            right = mid - 1;
        } else {
            // M개보다 많이 필요함 → 용량이 부족 → 키워야 함
            left = mid + 1;
        }
    }
    return result;
}

// compare 함수
static int compare(int capacity) {
    int count = 1;
    int sum = 0;

    for (int len : A) {
        if (sum + len > capacity) {
            count++;
            sum = len;
        } else {
            sum += len;
        }
    }
    return count - M;
}
}

```

```

9 3
1 2 3 4 5 6 7 8 9
17

```

3개의 블루레이에 강의를 순서대로 넣으면서, 가장 적은 용량으로 성공하려면 최소 용량은 17이어야 함

2343

goeun14109

모든 언어

▼

모든 결과

▼

검색

제출 번호	아이디	문제	결과	메모리	시간	언어	코드 길이	제출한 시간
94728893	goeun14109	2343	맞았습니다!!	111644 KB	848 ms	Java 11 / 수정	1187 B	20초 전