

Binary Search Basic

이분탐색 기초

競技プログラミングの鉄則

KPSC Algorithm Study 24/11/21 Thu.

by Haru_101

Search element on Array

- 이분탐색은 배열 내에서의 특정 값이나 어떤 조건을 만족하는 값을 효율적으로 찾는 방법
- 먼저, 크기가 N 인 배열에서 어떤 특정 값 x 가 존재하는지 찾는다고 가정합시다.
- 단순히 생각해 보면, 반복문 하나로 배열의 모든 값을 순회하면서 있는지 검사하면 될 것 같습니다.

Search element on Array

- 하지만, 배열의 크기가 크고 찾을 값이 여러개라면?
- 배열의 크기가 N 이고, 찾을 값의 개수(= 쿼리의 개수)가 Q 라고 합시다.
- 그러면, 한 번 값을 찾는 데에 최대 $O(N)$ 의 시간이 걸리고, 이를 Q 번 하므로, $O(QN)$ 이 걸립니다.
- $N = Q = 100,000$ 이면 시간 내에 문제를 해결하지 못합니다.

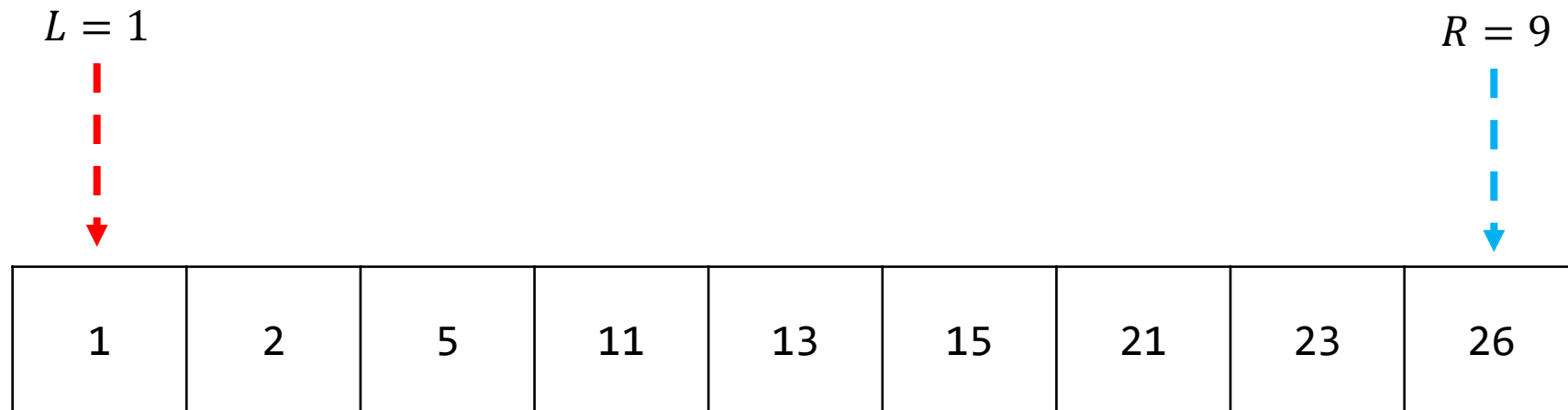
Search element on Array

- 이제 이분탐색(Binary Search)에 대해 배워봅시다.
- 이분탐색은 탐색의 범위를 계속 절반으로 줄여나가면서 값을 찾는 과정입니다. 그래서 이분탐색입니다.
- 배열이 다음과 같이 오름차순으로 정렬되어 있을 때, 23이라는 값을 찾겠다고 해봅시다.

1	2	5	11	13	15	21	23	26
---	---	---	----	----	----	----	----	----

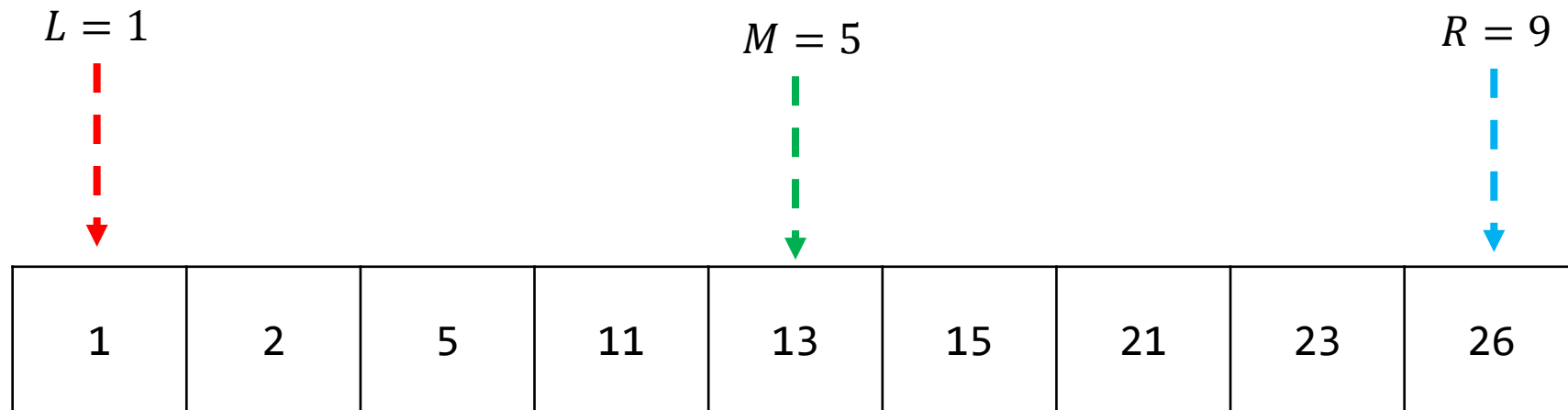
Search element on Array

- 먼저, 가장 왼쪽을 L , 가장 오른쪽을 R 로 잡습니다.
 - 편의상 1-indexed라고 합시다.



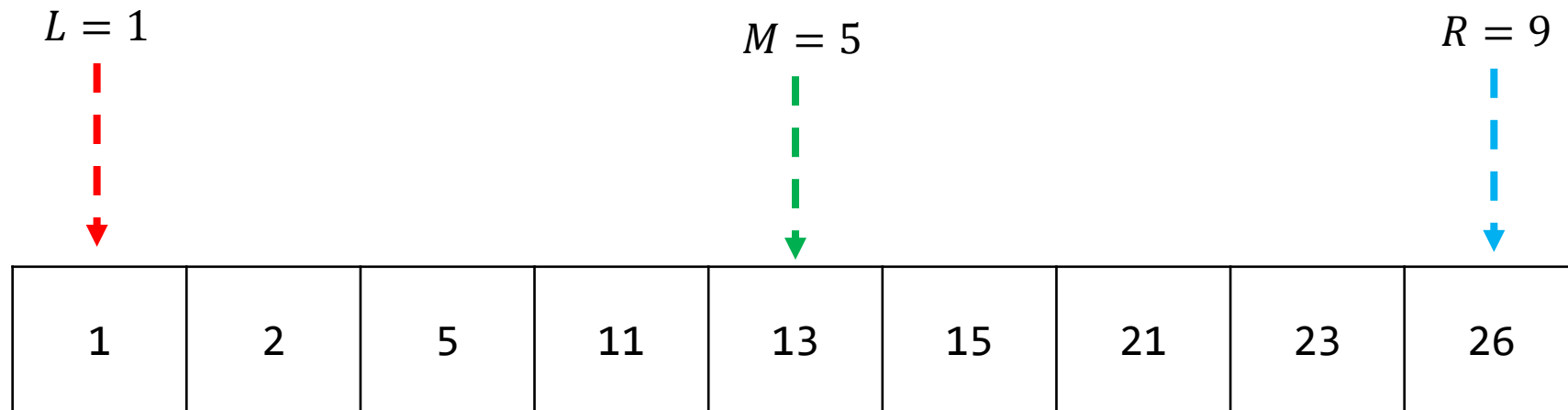
Search element on Array

- 그 다음, 중간을 나타내는 M 값을 잡습니다.
 - $M = \frac{(L+R)}{2}$ 이고, 소숫점 이하는 버립니다.



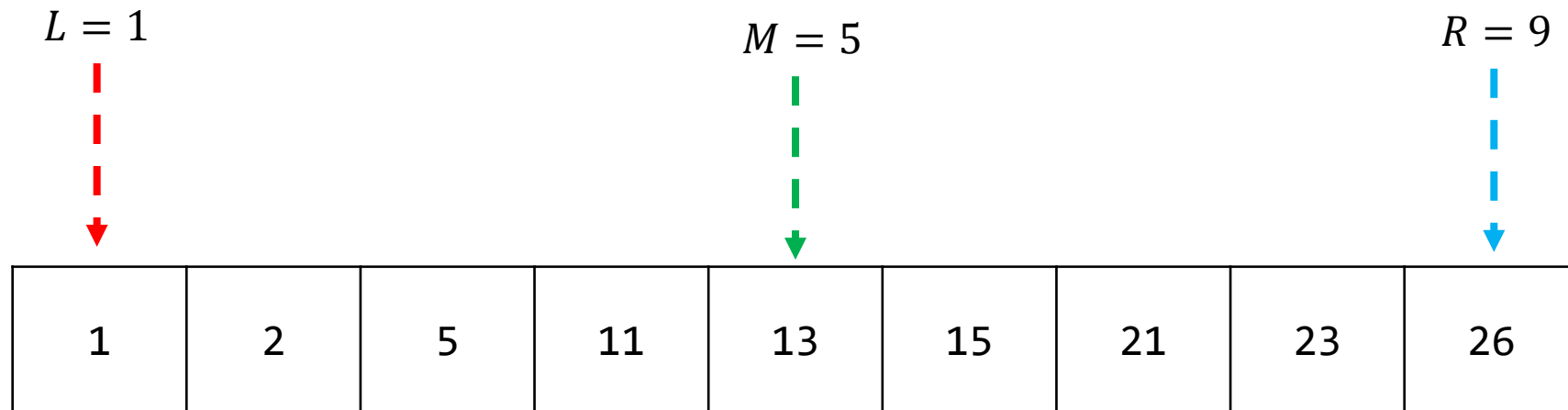
Search element on Array

- 이제, $M = 5$ 에 있는 값을 23과 비교해봅시다.
 - 13은 23보다 작네요.



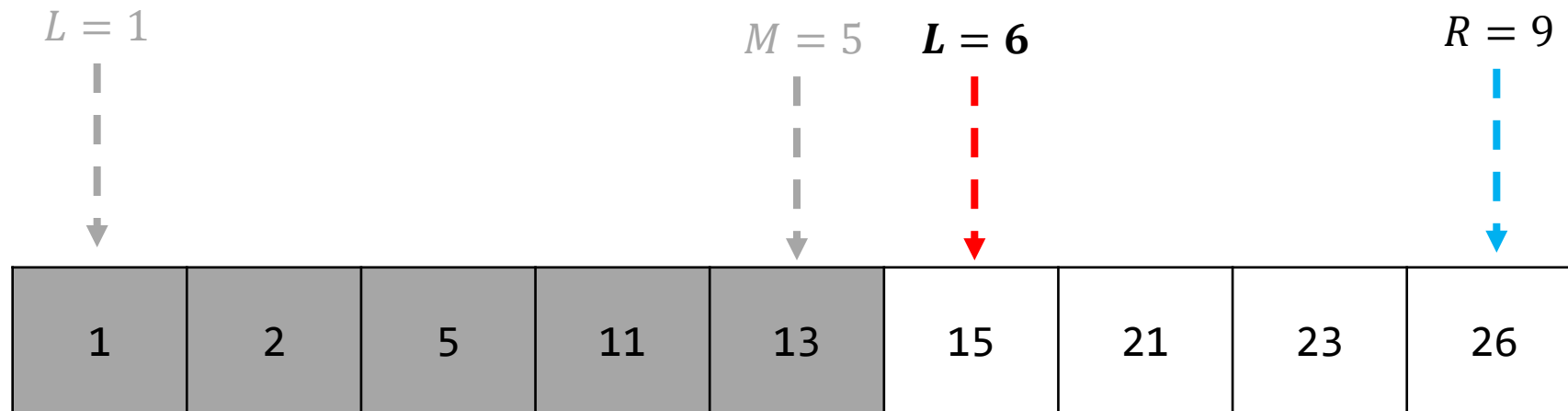
Search element on Array

- 그러면, 다음과 같이 결론을 내릴 수 있습니다.
 - $[L, M]$ 에 있는 값들은 우리가 찾고자 하는 값보다 작네.
 - 그러면, 더 이상 이 범위에서 23을 찾을 필요는 없네.



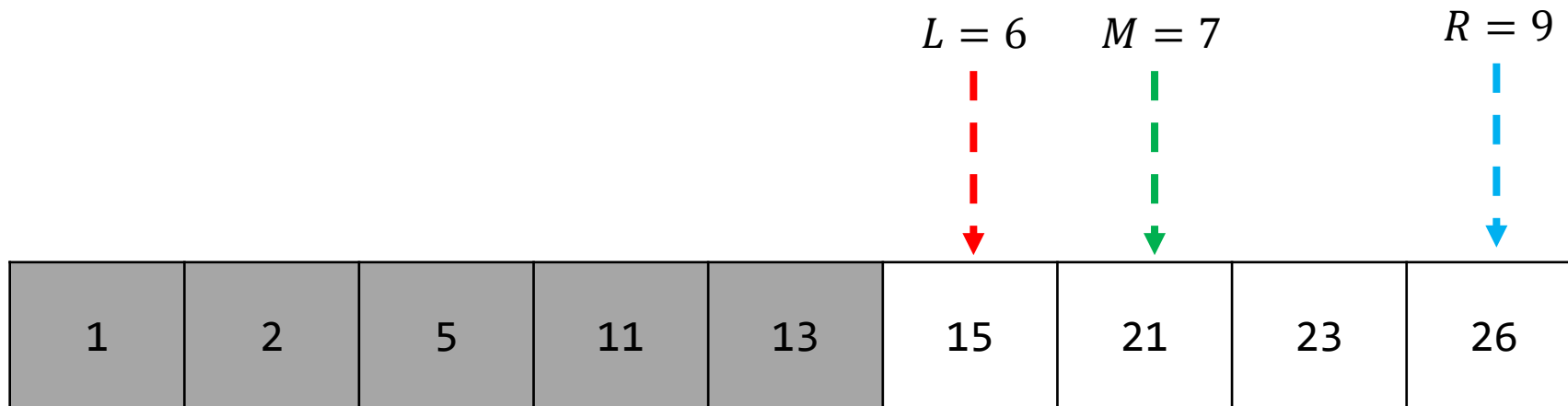
Search element on Array

- 따라서, 다음 번에 탐색할 범위를 줄입니다.
 - $[L, M]$ 에 우리가 찾고자 하는 값이 없으므로, 다음 번에 탐색할 범위를 $[M + 1, R]$ 로 축소!
 - $M + 1$ 인 이유는, M 에 찾고자 하는 값이 없기 때문입니다.



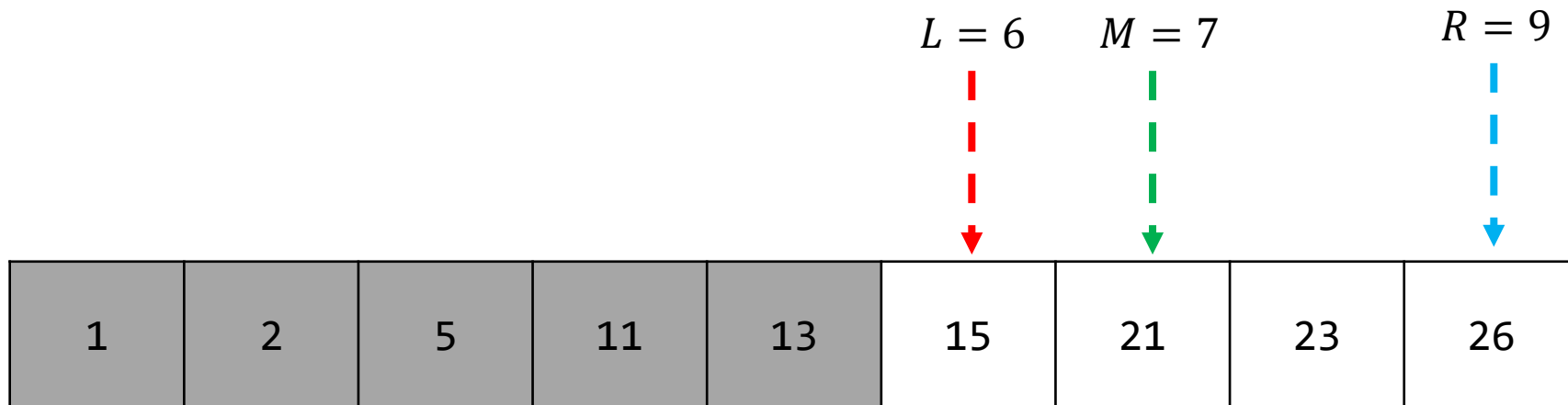
Search element on Array

- 지금까지 해왔던 과정을 다시 반복해봅시다.
 - 먼저 $M = \frac{L+R}{2}$ 값을 잡습니다.



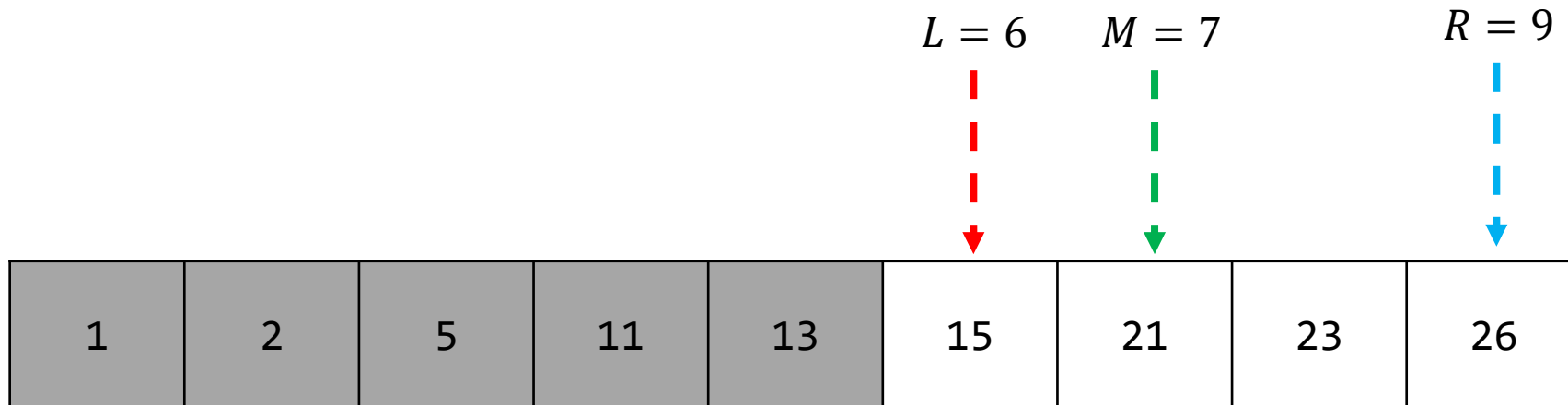
Search element on Array

- 이제, $M = 7$ 에 있는 값을 23과 비교해봅시다.
 - 21은 23보다 작네요.



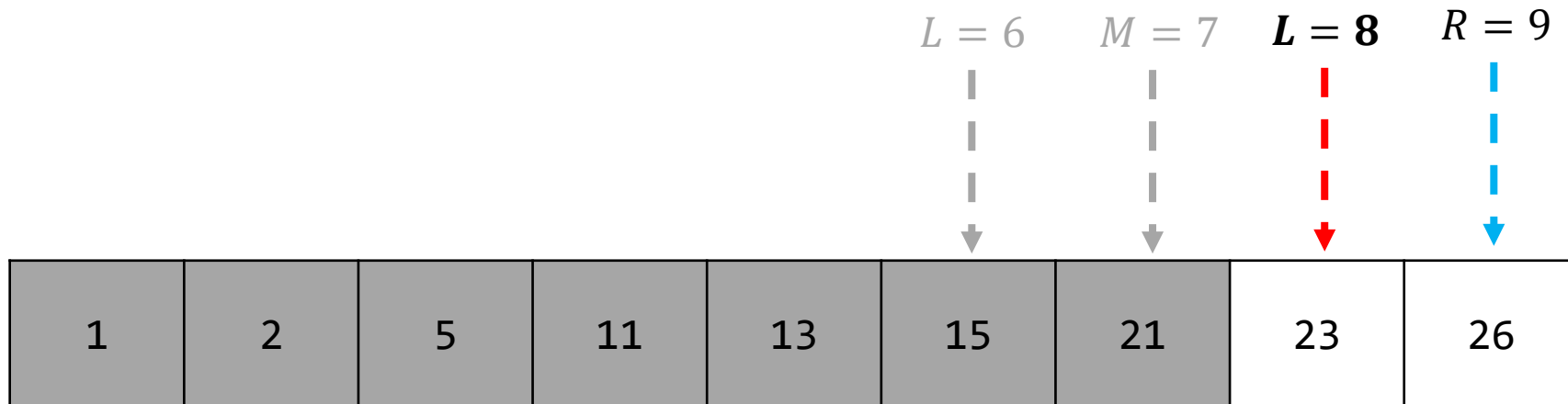
Search element on Array

- 그러면, 다음과 같이 결론을 내릴 수 있습니다.
 - $[L, M]$ 에 있는 값들은 우리가 찾고자 하는 값보다 작네.
 - 그러면, 더 이상 이 범위에서 23을 찾을 필요는 없네.



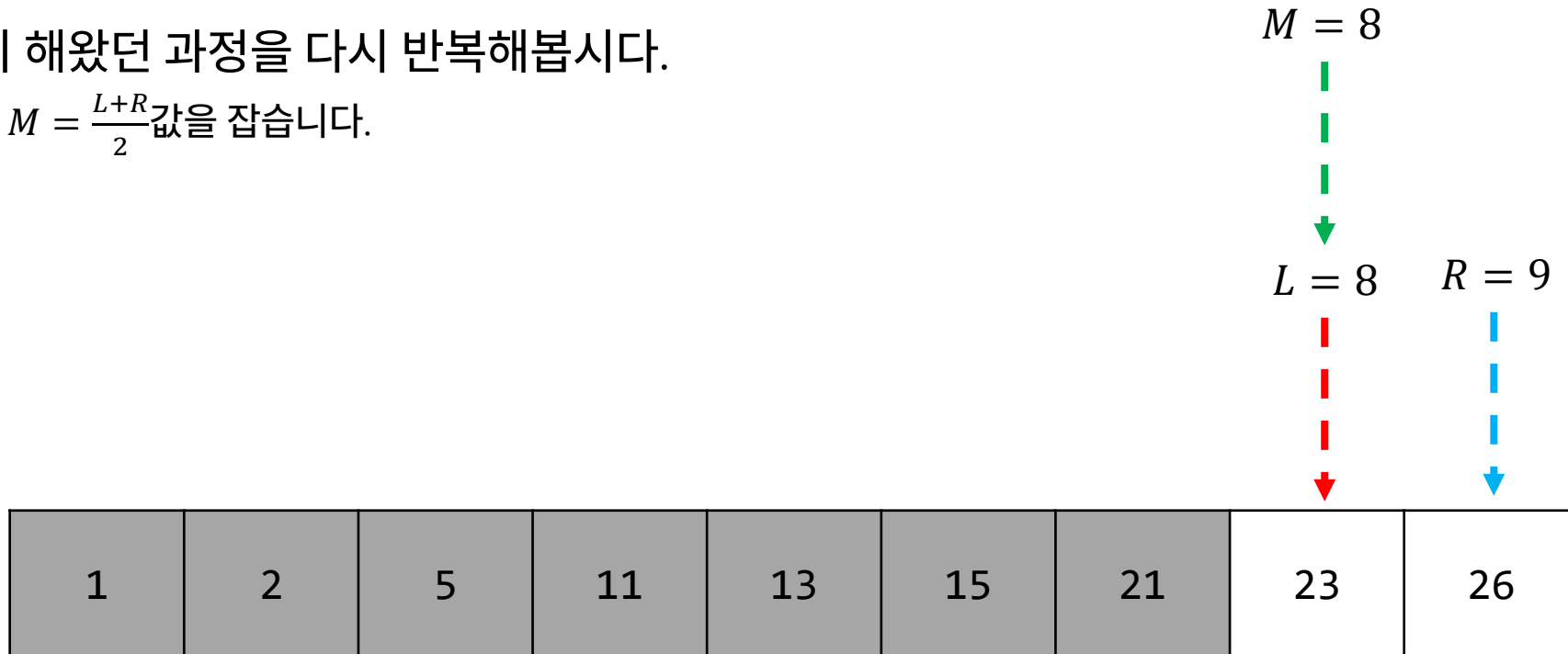
Search element on Array

- 따라서, 다음 번에 탐색할 범위를 줄입니다.
 - $[L, M]$ 에 우리가 찾고자 하는 값이 없으므로, 다음 번에 탐색할 범위를 $[M + 1, R]$ 로 축소!
 - $M + 1$ 인 이유는, M 에 찾고자 하는 값이 없기 때문입니다.



Search element on Array

- 지금까지 해왔던 과정을 다시 반복해봅시다.
 - 먼저 $M = \frac{L+R}{2}$ 값을 잡습니다.



Search element on Array

- 이제, $M = 8$ 에 있는 값을 23과 비교해봅시다.
 - 우리가 찾던 23이 M 에 있네요!



Search element on Array

- 우리가 찾고자 했던 23이 배열 내에 있으므로, 인덱스를 출력하거나 True등을 출력하면 됩니다.

1	2	5	11	13	15	21	23	26
---	---	---	----	----	----	----	----	----

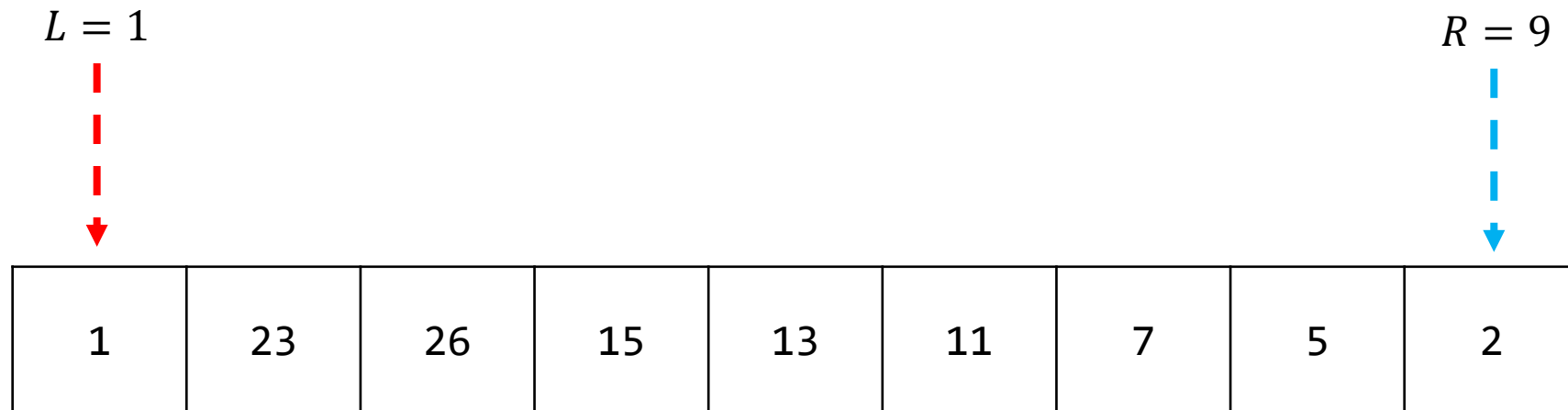
Search element on Array

- 근데, 이 배열이 정렬되어있지 않다면 어떻게 될까요?

1	23	26	15	13	11	7	5	2
---	----	----	----	----	----	---	---	---

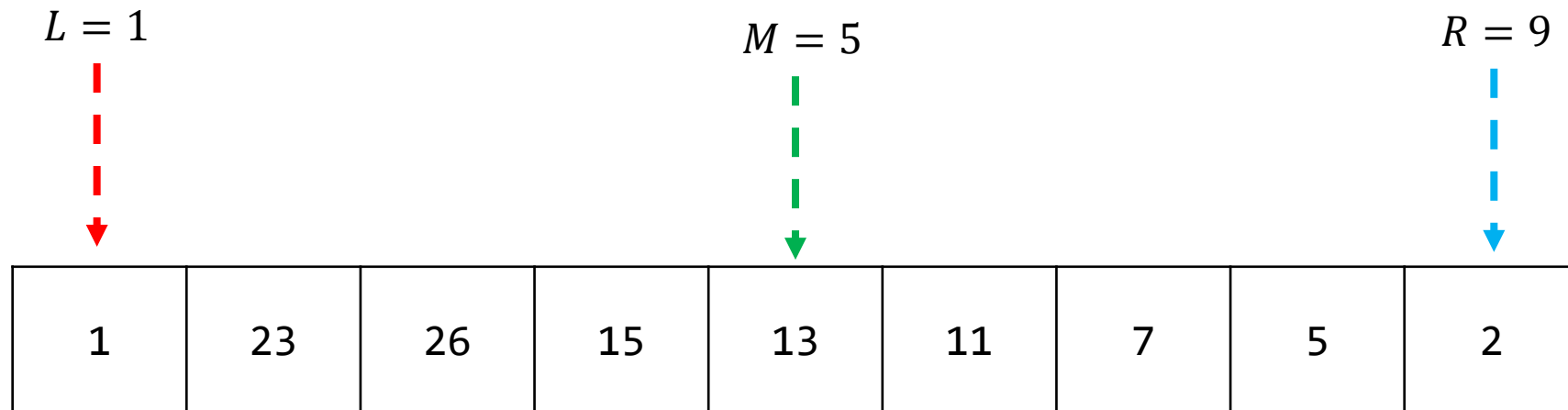
Search element on Array

- 먼저, 가장 왼쪽을 L , 가장 오른쪽을 R 로 잡습니다.



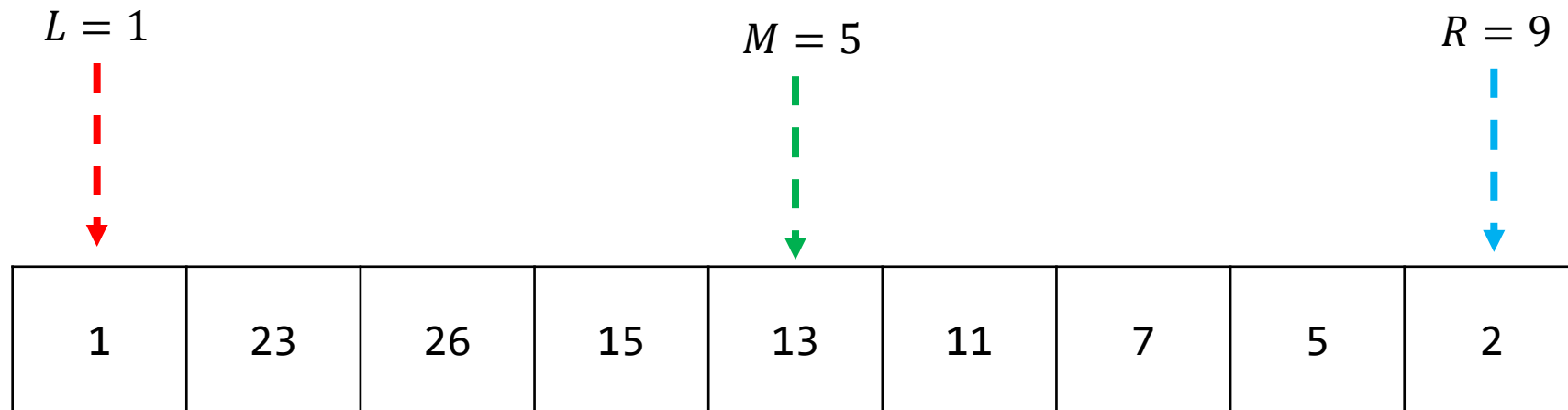
Search element on Array

- 그 다음, 중간을 나타내는 M 값을 잡습니다.
 - $M = \frac{(L+R)}{2}$ 이고, 소숫점 이하는 버립니다.



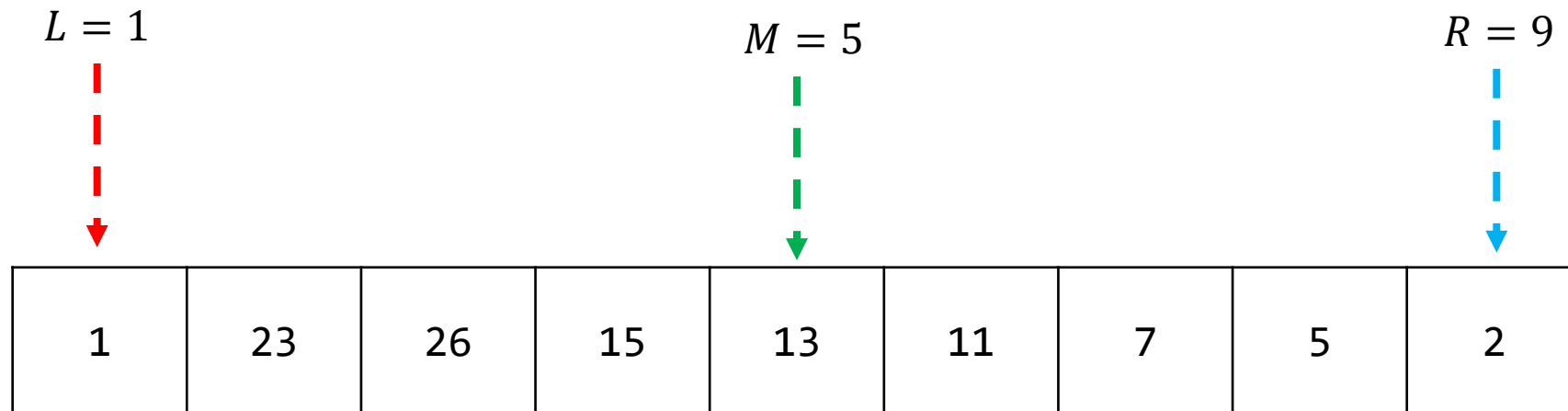
Search element on Array

- 이제, $M = 5$ 에 있는 값을 23과 비교해봅시다.
 - 13은 21보다 작네요.



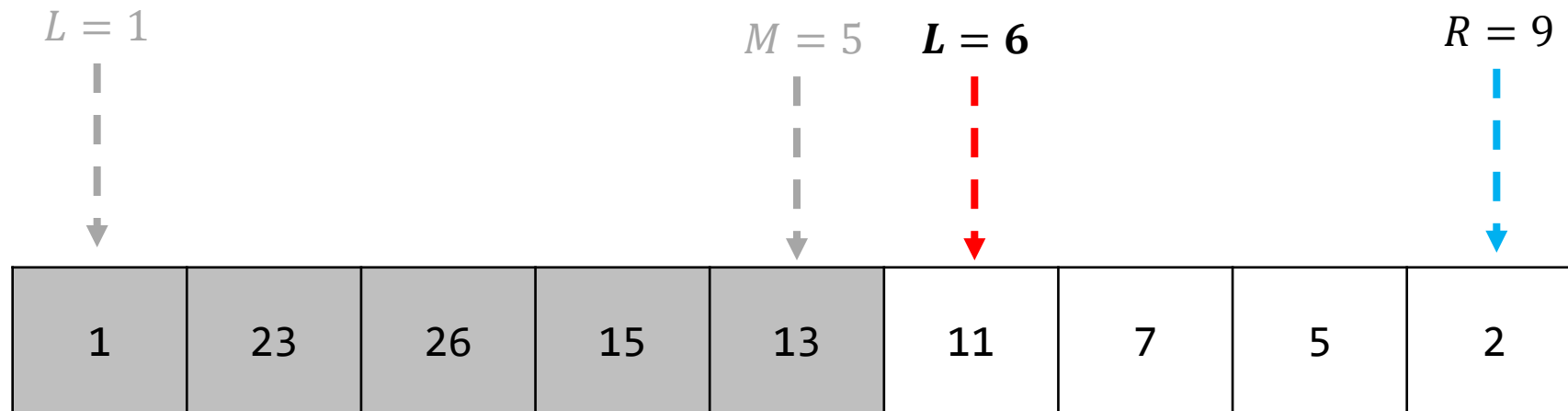
Search element on Array

- 그러면, 다음과 같이 결론을 내릴 수 있습니다.
 - $[L, M]$ 에 있는 값들은 우리가 찾고자 하는 값보다 작네.
 - 그러면, 더 이상 이 범위에서 23을 찾을 필요는 없네.



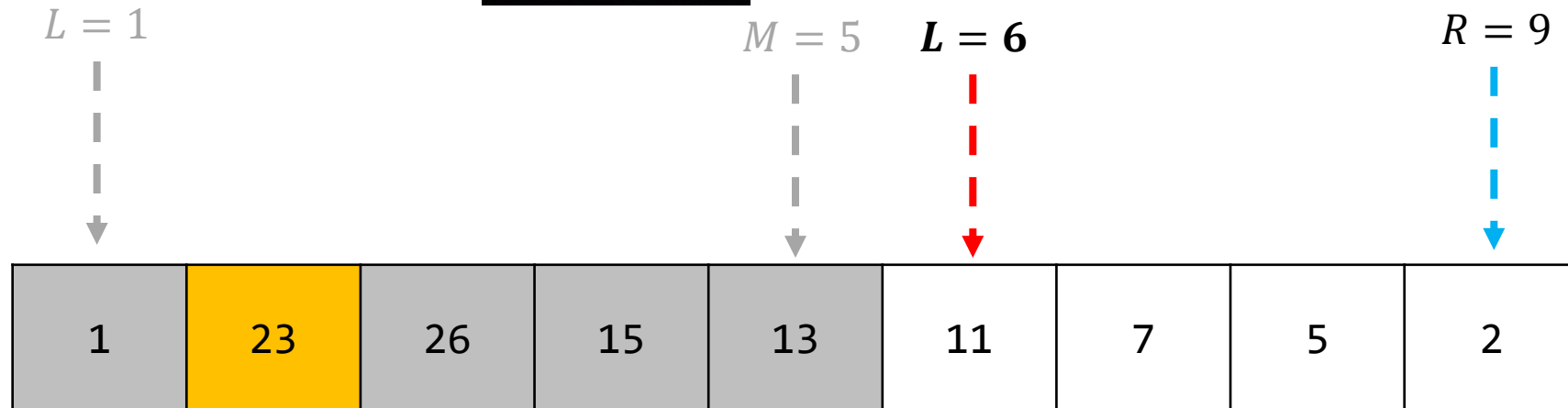
Search element on Array

- 따라서, 다음 번에 탐색할 범위를 줄입니다.
 - $[L, M]$ 에 우리가 찾고자 하는 값이 없으므로, 다음 번에 탐색할 범위를 $[M + 1, R]$ 로 축소
 - $M + 1$ 인 이유는, M 에 찾고자 하는 값이 없기 때문입니다.



Search element on Array

- 근데, 23이 회색 부분에 있네요?



Search element on Array

- 이런 일을 방지하고자, 찾고자 하는 배열, 범위에서는 단조성이 보장되어야 합니다.
- 단조성이란, 단조증가 혹은 단조감소를 의미합니다.
- 단조 증가 : $a < b$ 이면 $f(a) \leq f(b)$
- 단조 감소 : $a < b$ 이면 $f(a) \geq f(b)$

Search element on Array

- 문제를 풀어봅시다.
- https://atcoder.jp/contests/tessoku-book/tasks/tessoku_book_k

問題文

小さい順に並べられている、要素数 N の配列 $A = [A_1, A_2, \dots, A_N]$ があります。要素 X は配列 A の何番目に存在するかを出力してください。

なお、この問題は単純な全探索（→1.2節）でも解けますが、ここでは二分探索法を使って実装してください。

制約

- $1 \leq N \leq 100000$
- $1 \leq A_1 < A_2 < \dots < A_N \leq 10^9$
- 整数 X は A_1, A_2, \dots, A_N のいずれかである
- 오름차순으로 정렬된 크기가 N 인 배열 $A = [A_1, A_2, \dots, A_N]$ 이 있습니다. 원소 X 가 배열 A 의 몇 번째에 존재하는지 출력하세요.
- [제약] $X \in A$

Search element on Array

- 이분탐색 기본 구현이므로 따로 구체적인 설명은 스킵하고, 코드를 첨부합니다.

```

1 int arr[100005];
2 int main() {
3     fastio();
4     int N, X;
5     cin >> N >> X;
6     for(int i=1; i<=N; i++) {
7         cin >> arr[i];
8     }
9     int L = 1;
10    int R = N;
11    while(L <= R) {
12        int M = (L+R)/2;
13        if(arr[M] == X) {
14            cout << M;
15            break;
16        } else if(arr[M] < X) {
17            L = M+1;
18        } else {
19            R = M-1;
20        }
21    }
22 }

```

Search element on Array

- C++에서는 `lower_bound(start, end, val)` 을 이용해서 풀 수도 있습니다.

```

1 int arr[1000005];
2 int main() {
3     fastio();
4     int N, X;
5     cin >> N >> X;
6     for(int i=1; i<=N; i++) {
7         cin >> arr[i];
8     }
9     int idx = lower_bound(arr+1, arr+1+N, X) - arr;
10    if(idx <= N && arr[idx] == X) cout << idx;
11    else cout << -1;
12 }

```

Search element on Array

- 참고로, 앞 문제에선 배열이 이미 정렬된 순으로 주어져 있기 때문에 따로 정렬하지 않아도 됩니다.

Binary Search on a Condition

- 이분탐색은 배열 내에서 값을 찾는 것 뿐만 아니라 특정 조건을 만족하는 값을 찾는 데에도 유용합니다.
- 이 부분은 문제를 보면서 해설하겠습니다.

Binary Search on a Condition

- 문제를 풀어봅시다.
- https://atcoder.jp/contests/tessoku-book/tasks/tessoku_book_1

問題文

N 台のプリンターがあり、1 から N までの番号が付けられています。プリンター i は A_i 秒ごとにチラシを 1 枚印刷します。すなわち、スイッチを入れてから A_i 秒後、 $2A_i$ 秒後、 $3A_i$ 秒後... に印刷します。すべてのプリンターのスイッチを同時に入れたとき、 K 枚目のチラシが印刷されるのは何秒後でしょうか。

制約

- $1 \leq N \leq 100\,000$
 - $1 \leq K \leq 10^9$
 - $1 \leq A_i \leq 10^9$
 - 答えは 10^9 を超えない
 - 入力はすべて整数
- N 대의 프린터기가 있고, 각 프린터는 1부터 N 까지의 번호가 부여되어 있습니다. 프린터 i 는 A_i 초마다 전단지를 1장 인쇄합니다.
 - 다시 말하면, 전원을 키면 A_i 초 후, $2A_i$ 초 후, $3A_i$ 초 후, ...에 인쇄됩니다. 모든 프린터의 전원을 동시에 켤 때, 몇 초 뒤에 K 장 짜의 전단지가 인쇄되는 지 출력하시오.
 - [제약] 정답은 10^9 를 넘지 않음

Binary Search on a Condition

- 아래 예제에 대해서 살펴 봅시다.

4 10
1 2 3 4

- 그렇다면 인쇄 현황을 아래의 표로 나타낼 수 있습니다. (O는 해당 초에 인쇄됨을 의미함)

프린터 초	1	2	3	4	5	6	7	8	9
1	0	0	0	0	0	0	0	0	0
2		0		0		0		0	
3			0			0			0
4				0				0	

Binary Search on a Condition

- 일단, 각 프린터는 A_i 초마다 1장 인쇄하므로, 어떤 X 초까지 해당 프린터가 인쇄한 전단지의 장 수 α_i 는,

$$\alpha_i = \frac{X}{A_i}, \text{ 소수점 이하 버림}$$

- 위 식으로 나타낼 수 있습니다.

프린터 초	1	2	3	4	5	6	7	8	9
1	0	0	0	0	0	0	0	0	0
2		0		0		0		0	
3			0			0			0
4				0				0	

Binary Search on a Condition

- 예를 들어 프린터 2가 $A_2 = 2$ 초마다 1장 씩 인쇄하므로, $X = 7$ 초 후에 인쇄한 전단지의 장 수 α_2 는 $\alpha_2 = \frac{7}{2} = 3$ 이 됩니다. (소수점 버림)

프린터 초	1	2	3	4	5	6	7	8	9
1	0	0	0	0	0	0	0	0	0
2		0		0		0		0	
3			0			0			0
4				0				0	

Binary Search on a Condition

- 따라서, 모든 프린터에 대해 X 초 후에 인쇄한 전단지의 총 합은, $\sum \alpha_i$ 가 됩니다.
- 이를 $S = \sum \alpha_i$ 로 두고, 이 S 를 이용해 이분탐색을 하면 됩니다.

프린터 초	1	2	3	4	5	6	7	8	9
1	0	0	0	0	0	0	0	0	0
2		0		0		0		0	
3			0			0			0
4				0				0	

Binary Search on a Condition

- 'S로 이분탐색을 한다'가 감이 안오실 수 있을 것 같아서, 입력 예시를 가지고 설명을 하겠습니다.

4 10
1 2 3 4

- 먼저 $S \geq 10$ 인 S를 찾아야 합니다.
 - $S \leq 9$ 라면 10번째 전단지가 인쇄가 되지 않았음을 의미하기 때문입니다.

프린터 초	1	2	3	4	5	6	7	8	9
1	0	0	0	0	0	0	0	0	0
2		0		0		0		0	
3			0			0			0
4				0				0	

Binary Search on a Condition

- 이제 1초를 L , 9초를 R 로 잡습니다.
 - 참고 : 9초로 잡으면 안되지만, 이 입력 예제에서는 가능하므로 9초로 잡습니다. 추후 설명하겠습니다.

$L = 1$






$R = 9$



프린터 초	1	2	3	4	5	6	7	8	9
1	0	0	0	0	0	0	0	0	0
2		0		0		0		0	
3			0			0			0
4				0				0	

Binary Search on a Condition

- 그 다음, L, R 의 중간인 $M = \frac{1+9}{2} = 5$ 를 잡습니다.

	$L = 1$				$M = 5$				$R = 9$
									
프린터 초	1	2	3	4	5	6	7	8	9
1	0	0	0	0	0	0	0	0	0
2		0		0		0		0	
3			0			0			0
4				0				0	




Binary Search on a Condition

- 이제 $X = M$ 일 때 즉, X 초가 경과하였을 때 모든 프린터에서 전단지가 몇 장 나왔는지 계산해봅시다.

	<div> <div>$L = 1$</div> <div>$M = 5$</div> <div>$R = 9$</div> </div>								
프린터 초	1	2	3	4	5	6	7	8	9
1	0	0	0	0	0	0	0	0	0
2		0		0		0		0	
3			0			0			0
4				0				0	




Binary Search on a Condition

- 이제 $X = M$ 일 때 즉, X 초가 경과하였을 때 모든 프린터에서 전단지가 몇 장 나왔는지 계산해봅시다.

		$L = 1$			$M = 5$				$R = 9$	
										
	프린터 초	1	2	3	4	5	6	7	8	9
$\alpha_1 = 5$	1	0	0	0	0	0	0	0	0	0
$\alpha_2 = 2$	2		0		0		0		0	
$\alpha_3 = 1$	3			0			0			0
$\alpha_4 = 1$	4				0				0	
$S = 9$										

Binary Search on a Condition

- 지금 상황에선 $S = 9$ 입니다. 이 값은 $K = 10$ 보다 작으므로, $[L, M]$ 구간에는 찾고자 하는 답이 없음을 알 수 있습니다.

		$L = 1$			$M = 5$				$R = 9$	
										
	프린터 초	1	2	3	4	5	6	7	8	9
$\alpha_1 = 5$	1	0	0	0	0	0	0	0	0	0
$\alpha_2 = 2$	2		0		0		0		0	
$\alpha_3 = 1$	3			0			0			0
$\alpha_4 = 1$	4				0				0	
$S = 9$										

Binary Search on a Condition

- 따라서, L 을 $M + 1$ 로 옮기고, 다시 이 과정을 반복해봅시다.

	$L = 1$				$M = 5$	$L = 6$			$R = 9$
	↓				↓	↓			↓
프린터 초	1	2	3	4	5	6	7	8	9
1	0	0	0	0	0	0	0	0	0
2		0		0		0		0	
3			0			0			0
4				0				0	




Binary Search on a Condition

- L, R 의 중간인 $M = \frac{6+9}{2} = 7$ 을 잡습니다.

프린터 초						$L = 6$	$M = 7$		$R = 9$
	1	2	3	4	5	6	7	8	9
1	0	0	0	0	0	0	0	0	0
2		0		0		0		0	
3			0			0			0
4				0				0	




Binary Search on a Condition

- 이제 $X = M$ 일 때 즉, X 초가 경과하였을 때 모든 프린터에서 전단지가 몇 장 나왔는지 계산해봅시다.

						$L = 6$	$M = 7$	$R = 9$	
									
프린터 초	1	2	3	4	5	6	7	8	9
1	0	0	0	0	0	0	0	0	0
2		0		0		0		0	
3			0			0			0
4				0				0	

Binary Search on a Condition

- 이제 $X = M$ 일 때 즉, X 초가 경과하였을 때 모든 프린터에서 전단지가 몇 장 나왔는지 계산해봅시다.

							$L = 6$	$M = 7$	$R = 9$	
										
프린터 초		1	2	3	4	5	6	7	8	9
$\alpha_1 = 7$	1	0	0	0	0	0	0	0	0	0
$\alpha_2 = 3$	2		0		0		0		0	
$\alpha_3 = 2$	3			0			0			0
$\alpha_4 = 1$	4				0				0	

$S = 13$

Binary Search on a Condition

- 지금 상황에선 $S = 13$ 입니다. 이 값은 $K = 10$ 보다 크므로, $[M, R]$ 구간에서도 조건을 만족한다는 것은 알 수 있습니다.
- 일단 이 구간에서 조건을 가장 먼저 만족하는 시간이 M 이므로, $L = 6$ $M = 7$ $R = 9$
 $M = 7$ 값을 정답으로 임시로 저장해둡니다.

	프린터 초	1	2	3	4	5	6	7	8	9
$\alpha_1 = 7$	1	0	0	0	0	0	0	0	0	0
$\alpha_2 = 3$	2		0		0		0		0	
$\alpha_3 = 2$	3			0			0			0
$\alpha_4 = 1$	4				0				0	

$S = 13$

Binary Search on a Condition

- 문제는, $M = 7$ 이전에 이미 K 번째 전단지가 인쇄되었을 수 있습니다.
- 즉, 6초 때, $K = 10$ 번째 전단지가 인쇄되었을 수 있습니다.
- 따라서 정답을 확정해두지 않고 저장만 해둡니다.

$L = 6$

$M = 7$

$R = 9$

	프린터 초	1	2	3	4	5	6	7	8	9
$\alpha_1 = 7$	1	0	0	0	0	0	0	0	0	0
$\alpha_2 = 3$	2		0		0		0		0	
$\alpha_3 = 2$	3			0			0			0
$\alpha_4 = 1$	4				0				0	

$S = 13$

Binary Search on a Condition

- 지금 상황에선 $S = 13$ 입니다. 이 값은 $K = 10$ 보다 크므로, $[M, R]$ 구간에는 더 이상 최적의 해가 존재하지 않습니다. (M 제외) 따라서, R 을 $M - 1$ 로 변경하고 지금까지 해왔던 과정을 반복하면 됩니다.

$L = 6$

$M = 7$

$R = 9$

	프린터 초	1	2	3	4	5	6	7	8	9
$\alpha_1 = 7$	1	0	0	0	0	0	0	0	0	0
$\alpha_2 = 3$	2		0		0		0		0	
$\alpha_3 = 2$	3			0			0			0
$\alpha_4 = 1$	4				0				0	

$S = 13$

Binary Search on a Condition

- L, R 의 중간인 $M = \frac{6+6}{2} = 6$ 을 잡습니다.

$$L = M = R = 6$$



프린터 초	1	2	3	4	5	6	7	8	9
1	0	0	0	0	0	0	0	0	0
2		0		0		0		0	
3			0			0			0
4				0				0	

Binary Search on a Condition

- 이제 $X = M$ 일 때 즉, X 초가 경과하였을 때 모든 프린터에서 전단지가 몇 장 나왔는지 계산해봅시다.

$$L = M = R = 6$$



프린터 초	1	2	3	4	5	6	7	8	9
$\alpha_1 = 6$	1	0	0	0	0	0	0	0	0
$\alpha_2 = 3$	2		0		0	0		0	
$\alpha_3 = 2$	3			0		0			0
$\alpha_4 = 1$	4				0			0	

$$S = 12$$

Binary Search on a Condition

- 지금 상황에선 $S = 12$ 입니다. 이 값은 $K = 10$ 보다 큼니다.
- 아까 저장했던 정답 후보가 $M = 7$ 인데, $M = 6$ 에서도 조건을 만족하므로 정답 후보를 다시 6으로 바꿉니다.

$$L = M = R = 6$$



프린터 초	1	2	3	4	5	6	7	8	9
$\alpha_1 = 6$	1	0	0	0	0	0	0	0	0
$\alpha_2 = 3$	2	0		0		0		0	
$\alpha_3 = 2$	3		0			0			0
$\alpha_4 = 1$	4			0				0	

$S = 12$

Binary Search on a Condition

- 이제 $L = R$ 이므로 더 이상 반복문이 돌지 않기 때문에 반복이 종료되고, 최종적으로 정답은 6이 됩니다.

$$L = M = R = 6$$



프린터 초	1	2	3	4	5	6	7	8	9
$\alpha_1 = 6$	1	0	0	0	0	0	0	0	0
$\alpha_2 = 3$	2		0		0	0		0	
$\alpha_3 = 2$	3			0		0			0
$\alpha_4 = 1$	4				0			0	

$$S = 12$$

Binary Search on a Condition

- [단조성 증명]
- 이전에 이분탐색을 하려면 단조성을 만족해야 한다고 언급했었습니다.
- 이 문제가 단조성을 만족하는지 간단하게 봐봅시다.

Binary Search on a Condition

- 먼저, $t = i$ 에 인쇄되는 전단지의 수는 0장 이상입니다. 이 값을 α 라고 하면,
- $S_{t=i-1} + \alpha = S_{t=i}$ 가 됩니다.
- 여기서, α 에 대해 식을 정리하면, $\alpha = S_{t=i} - S_{t=i-1}$ 이 되고, $\alpha \geq 0$ 이므로 $S_{t=i} - S_{t=i-1} \geq 0$ 입니다.
- 따라서, $S_{t=i} \geq S_{t=i-1}$ 이므로, t 초까지 인쇄된 전단지 수의 합을 $f(t)$ 라고 할 때, $f(t)$ 는 단조증가 함수입니다.
- 물론 직관적으로 볼 때 이게 보일 수 있지만, 식으로 한 번 정리해보는 것도 나쁘진 않은 것 같습니다.

Challenge

- 문제를 풀어봅시다.
- https://atcoder.jp/contests/tessoku-book/tasks/tessoku_book_n

問題文

A・B・C・Dの4つの箱があります。各箱には、以下の N 枚のカードが入っています。

- 箱Aには整数 A_1, A_2, \dots, A_N が書かれたカードがある。
- 箱Bには整数 B_1, B_2, \dots, B_N が書かれたカードがある。
- 箱Cには整数 C_1, C_2, \dots, C_N が書かれたカードがある。
- 箱Dには整数 D_1, D_2, \dots, D_N が書かれたカードがある。

あなたはそれぞれの箱から1枚ずつカードを取り出します。取り出した4枚のカードに書かれた整数の合計が K となる可能性はあるか、判定してください。

制約

- $1 \leq N \leq 1000$
 - $1 \leq K \leq 10^8$
 - $1 \leq A_x, B_y, C_z, D_w \leq 10^8$
- A, B, C, D 4개의 상자에는 각각 N 개의 카드가 있고, A 상자에는 A_1, \dots, A_N , B 상자에는 B_1, \dots, B_N , C 상자에는 C_1, \dots, C_N , D 상자에는 D_1, \dots, D_N 이 써져있는 카드가 있습니다.
 - 각 상자에서 카드 한 장을 뽑습니다. 이때 뽑은 카드 4장의 합이 정확히 K 가 될 수 있는지 출력하는 프로그램을 작성하시오.

Challenge

- 문제를 단순히 4중 반복문으로 풀고자 하니... $N = 10^3$ 이라서 $O(N^4)$ 은 당연히 안 될 것 같습니다.
- 어떻게 풀어야 할까요?

Challenge

- 결론부터 말하면 $A_i + B_j$ 를 다 나열한 배열 AB 와 $C_i + D_j$ 를 다 나열한 배열 CD 를 가지고 문제를 풀 수 있습니다.
- 즉, $A_i + B_j + C_k + D_l = K$ 를 만족하는 (i, j, k, l) 을 직접적으로 구하는 것이 아니라, $AB_i + CD_j = K$ 를 만족하는 (i, j) 를 찾으면 됩니다.
- 더 깊게 생각해보면 K 는 이미 문제에서 주어져 있고, AB_i 는 우리가 지정할 수 있는 값입니다.
- 따라서, CD_j 는 $K - AB_i$ 이므로, $K - AB_i$ 가 CD_j 가 배열 내에 존재하는 지 검사하면 됩니다.