

알고리즘 기초 세미나

최재혁

200522

이분 탐색 (Binary Search)

	Low = 0			Mid = 4			High = 9			
Target : 23	2	5	8	12	16	23	38	56	72	91
16 < 23 Low = Mid+1	2	5	8	12	16	23	38	56	72	91
23 < 56 High = Mid-1	2	5	8	12	16	23	38	56	72	91
				Low, Mid = 5			High = 6			
	2	5	8	12	16	23	38	56	72	91
				Low, Mid = 5			High = 6			
Return 5	2	5	8	12	16	23	38	56	72	91

이분 탐색 (Binary Search)

	Low = 0			Mid = 4			High = 9			
Target : 27	2	5	8	12	16	23	38	56	72	91
16 < 27 Low = Mid+1	2	5	8	12	16	23	38	56	72	91
27 < 56 High = Mid-1	2	5	8	12	16	23	38	56	72	91
23 < 27 Low = Mid+1	2	5	8	12	16	23	38	56	72	91
Return -1	2	5	8	12	16	23	38	56	72	91

구현

```
1 #include <iostream>
2 using namespace std;
3
4 int binarySearch(int arr[], int l, int r, int x)
5 {
6     while (l <= r) {
7         int m = l + (r - l) / 2;
8         if (arr[m] == x)
9             return m;
10        if (arr[m] < x)
11            l = m + 1;
12        else
13            r = m - 1;
14    }
15    return -1;
16 }
17
18 int main(void)
19 {
20     int arr[] = { 2, 5, 8, 12, 16, 23, 38, 56, 72, 91 }, x, n = 10;
21     cin >> x;
22
23     int result = binarySearch(arr, 0, n - 1, x);
24     if(result != -1)
25         cout << "arr[" << result << "] : " << x << "\n";
26     else
27         cout << x << " is not in array\n";
28     return 0;
29 }
```

10815번, 숫자 카드 (acmicpc.net/problem/10815)

숫자 카드

성공

분류



4 Silver IV

이문 탐색 정렬

시간 제한	메모리 제한	제출	정답	맞은 사람	정답 비율
2 초	256 MB	20490	10013	6759	48.542%

문제

숫자 카드는 정수 하나가 적혀져 있는 카드이다. 상근이는 숫자 카드 N 개를 가지고 있다. 정수 M 개가 주어졌을 때, 이 수가 적혀있는 숫자 카드를 상근이가 가지고 있는지 아닌지를 구하는 프로그램을 작성하시오.

입력

첫째 줄에 상근이가 가지고 있는 숫자 카드의 개수 N ($1 \leq N \leq 500,000$)이 주어진다. 둘째 줄에는 숫자 카드에 적혀있는 정수가 주어진다. 숫자 카드에 적혀있는 수는 $-10,000,000$ 보다 크거나 같고, $10,000,000$ 보다 작거나 같다. 두 숫자 카드에 같은 수가 적혀있는 경우는 없다.

셋째 줄에는 M ($1 \leq M \leq 500,000$)이 주어진다. 넷째 줄에는 상근이가 가지고 있는 숫자 카드인지 아닌지를 구해야 할 M 개의 정수가 주어지며, 이 수는 공백으로 구분되어져 있다. 이 수도 $-10,000,000$ 보다 크거나 같고, $10,000,000$ 보다 작거나 같다.

출력

첫째 줄에 입력으로 주어진 M 개의 수에 대해서, 각 수가 적힌 숫자 카드를 상근이가 가지고 있으면 1을, 아니면 0을 공백으로 구분해 출력한다.

예제 입력 1 복사

```
5
6 3 2 10 -10
8
10 9 -5 2 3 4 5 -10
```

예제 출력 1 복사

```
1 0 0 1 1 0 0 1
```

어떻게 풀까?

- 숫자 카드들을 모두 입력 받은 뒤 정렬
- 각 숫자가 들어올 때마다, 해당 숫자로 이분탐색

풀이 (10815.cpp)

```
1 #include <iostream>
2 #include <algorithm>
3 using namespace std;
4
5 bool binarySearch(int arr[], int l, int r, int x)
6 {
7     while (l <= r) {
8         int m = l + (r - l) / 2;
9         if (arr[m] == x)
10             return true;
11         if (arr[m] < x)
12             l = m + 1;
13         else
14             r = m - 1;
15     }
16     return false;
17 }
18
19 int main(void)
20 {
21     ios_base::sync_with_stdio(false);
22     cin.tie(NULL);
23
24     int n;
25     cin >> n;
26     int arr[n], m, t, ret;
27     for(int i = 0; i < n; i++)
28         cin >> arr[i];
29     cin >> m;
30     sort(arr, arr + n);
31
32     for(int i = 0; i < m; i++){
33         cin >> t;
34         ret = binarySearch(arr, 0, n-1, t);
35         if(ret)
36             cout << "1 ";
37         else
38             cout << "0 ";
39     }
40     return 0;
41 }
```

1654번, 랜선 자르기 (acmicpc.net/problem/1654)

랜선 자르기 성공 분류



Silver III

[이문 탐색](#)

시간 제한	메모리 제한	제출	정답	맞은 사람	정답 비율
2 초	128 MB	42619	8799	5576	19.601%

문제

집에서 시간을 보내던 오영식은 박성원의 부름을 받고 급히 달려왔다. 박성원이 캠프 때 쓸 N 개의 랜선을 만들어야 하는데 너무 바빠서 영식에게 도움을 청했다.

이미 오영식은 자체적으로 K 개의 랜선을 가지고 있다. 그러나 K 개의 랜선은 길이가 제각각이다. 박성원은 랜선을 모두 N 개의 같은 길이의 랜선으로 만들고 싶었기 때문에 K 개의 랜선을 잘라서 만들어야 한다. 예를 들어 300cm 짜리 랜선에서 140cm 짜리 랜선을 두 개 잘라내면 20cm 은 버려야 한다. (이미 자른 랜선은 붙일 수 없다.)

편의를 위해 랜선을 자르거나 만들 때 손실되는 길이는 없다고 가정하며, 기존의 K 개의 랜선으로 N 개의 랜선을 만들 수 없는 경우는 없다고 가정하자. 그리고 자를 때는 항상 센티미터 단위로 정수길이만큼 자른다고 가정하자. N 개보다 많이 만드는 것도 N 개를 만드는 것에 포함된다. 이때 만들 수 있는 최대 랜선의 길이를 구하는 프로그램을 작성하시오.

입력

첫째 줄에는 오영식이 이미 가지고 있는 랜선의 개수 K , 그리고 필요한 랜선의 개수 N 이 입력된다. K 는 1 이상 10,000이하의 정수이고, N 은 1 이상 1,000,000이하의 정수이다. 그리고 항상 $K \leq N$ 이다. 그 후 K 줄에 걸쳐 이미 가지고 있는 각 랜선의 길이가 센티미터 단위의 정수로 입력된다. 랜선의 길이는 $2^{31}-1$ 보다 작거나 같은 자연수이다.

출력

첫째 줄에 N 개를 만들 수 있는 랜선의 최대 길이를 센티미터 단위의 정수로 출력한다.

예제 입력 1 복사

```
4 11
802
743
457
539
```

예제 출력 1 복사

```
200
```


어떻게 풀까?

- 어떤 길이로 자르면 몇 개가 남을지를 이분 탐색으로 찾아본다

풀이 (10815.cpp)

```
1 #include <iostream>
2 using namespace std;
3
4 long long len[10000];
5 int main()
6 {
7     long long n, k, i, hi, lo, md, sm, ret = 0;
8     cin >> n >> k;
9     for (i = 0; i < n; i++)
10         cin >> len[i];
11
12     hi = 2147483647;
13     lo = 1;
14     while (hi >= lo)
15     {
16         md = (hi + lo) / 2;
17
18         for (sm = 0, i = 0; i < n; i++)
19             sm += len[i] / md;
20
21         if (sm >= k)
22         {
23             ret = ret < md ? md : ret;
24             lo = md + 1;
25         }
26         else
27         {
28             hi = md - 1;
29         }
30     }
31
32     cout << ret;
33
34     return 0;
35 }
36
```

잡다한 팁

- `#include <bits/stdc++.h>`
- GCC에서 한번에 모든 표준 헤더를 추가할 때 사용
- 필요 없는 것 까지 include하는 것이라 컴파일 시간이 늘지만
- 잡다한 헤더를 다 외우거나 쳐줄 필요가 없음

잡다한 팁

```
1 #include <stdio>
2 using namespace std;
3
4 int main(void)
5 {
6     int arr[1000], idx = 0;
7
8     while(~scanf("%d", &arr[idx]))
9         idx++;
10
11     for(int i = 0; i < idx; i++)
12         printf("%d ", arr[i]);
13
14     return 0;
15 }
```