

# 이분탐색: 수 찾기

#백준 1920번

## 문제

N개의 정수 A[1], A[2], ..., A[N]이 주어져 있을 때, 이 안에 X라는 정수가 존재하는지 알아내는 프로그램을 작성하시오.

## 입력

첫째 줄에 자연수 N( $1 \leq N \leq 100,000$ )이 주어진다. 다음 줄에는 N개의 정수 A[1], A[2], ..., A[N]이 주어진다. 다음 줄에는 M( $1 \leq M \leq 100,000$ )이 주어진다. 다음 줄에는 M개의 수들이 주어지는데, 이 수들이 A안에 존재하는지 알아내면 된다. 모든 정수의 범위는 -231 보다 크거나 같고 231보다 작다.

## 출력

M개의 줄에 답을 출력한다. 존재하면 1을, 존재하지 않으면 0을 출력한다.

## 예제 입력

```
5
4 1 5 2 3
5
1 3 7 9 5
```

## 예제 출력

```
1
1
0
0
1
```

## Try 1

- 이분탐색 함수 구현은 잘한듯
- **but**, 시간 초과

```
#include <iostream>
#include <vector>
#include <algorithm>
using namespace std;

int binarySearch(vector<int> A,int begin,int end, int key){

    while(begin<=end){
        int middle=(begin+end)/2;
```

```

        if(begin==end){
            if(key==A[begin]){
                return 1;
            }
            else{
                return 0;
            }
        }

        if(key==A[middle]){
            return 1;
        }
        else if (key<A[middle]){
            end=middle-1;
        }
        else if (key>A[middle]){
            begin=middle+1;
        }
        else{
            return 0;
        }
    }
}

int main()
{
    int N,M;
    cin>>N;
    vector<int> A;
    vector<int> B;
    bool answer;
    for(int i=0;i<N;i++){
        int tmp;
        cin>>tmp;
        A.push_back(tmp);
    }
    sort(A.begin(),A.end());

    cin>>M;
    for(int i=0;i<M;i++){
        int tmp;
        cin>>tmp;
        B.push_back(tmp);
    }

    for(int i=0;i<M;i++){

```

```

        answer=binarySearch(A,0,N-1,B[i]);
        if(answer){
            cout<<1<<endl;
        }
        else{
            cout<<0<<endl;
        }
    }
    return 0;
}

```

## Try 2

- [binary search 라이브러리 활용하기](#)
- [endl 대신 '\n' 사용하기](#)

: [endl](#) 은 출력 후 버퍼를 즉시 비우기 때문에, 대량의 데이터를 출력할 때 성능 저하 발생 가능

- [입출력 최적화](#)

- `ios::sync_with_stdio(false);`
- `cin.tie(NULL);`

→ 입출력 속도 향상, 대규모 입력 받을 때 유용

```

#include <iostream>
#include <vector>
#include <algorithm>
using namespace std;

int main()
{
    ios::sync_with_stdio(false); // 입출력 속도 향상
    cin.tie(NULL); // 입출력 동기화 해제

    int N,M;
    cin>>N;
    vector<int> A(N);

    for(int i=0;i<N;i++){
        cin>>A[i];
    }
    sort(A.begin(),A.end());

    cin>>M;
    for(int i=0;i<M;i++){
        int B; //배열에 굳이 담을 필요 없었음
        cin>>B; //바로 입력 받아서 확인하기
    }
}

```

```
        if (binary_search(A.begin(), A.end(), B)) { //라이브러리 사용!!  
            cout << 1 << '\n';  
        }  
        else {  
            cout << 0 << '\n';  
        }  
    }  
    return 0;  
}
```