

시간복잡도

이분탐색

발표자 : 이윤수(lys9546)

# 시간복잡도 왜 필요한가?

5 2751번

제출

맞힌 사람

숏코딩

재채점 결과

채점 현황

내 제출

🔗 난이도 기여

강의 ▾

질문 검색

수 정렬하기 2 

성공

☆

시간 제한	메모리 제한	제출	정답	맞힌 사람	정답 비율
2 초	256 MB	217280	62221	43221	30.441%

문제

N개의 수가 주어졌을 때, 이를 오름차순으로 정렬하는 프로그램을 작성하시오.

입력

첫째 줄에 수의 개수  $N(1 \leq N \leq 1,000,000)$ 이 주어진다. 둘째 줄부터 N개의 줄에는 수가 주어진다. 이 수는 절댓값이 1,000,000보다 작거나 같은 정수이다. 수는 중복되지 않는다.

출력

첫째 줄부터 N개의 줄에 오름차순으로 정렬한 결과를 한 줄에 하나씩 출력한다.

<https://www.acmicpc.net/problem/2751>

# 시간복잡도 왜 필요한가?

제출 번호	아이디	문제	결과	메모리	시간	언어	코드 길이	제출한 시간
12895054	lys9546	 2751	시간 초과			C99 / 수정	881 B	3년 전
12883531	lys9546	 2751	맞았습니다!!	12428 KB	484 ms	C99 / 수정	1098 B	3년 전
12883505	lys9546	 2751	틀렸습니다			C99 / 수정	1098 B	3년 전
12858303	lys9546	 2751	맞았습니다!!	4904 KB	528 ms	C99 / 수정	1555 B	3년 전
12858294	lys9546	 2751	틀렸습니다			C99 / 수정	1555 B	3년 전
12858277	lys9546	 2751	틀렸습니다			C99 / 수정	1555 B	3년 전
12858147	lys9546	 2751	시간 초과			C99 / 수정	1543 B	3년 전

# 시간복잡도 왜 필요한가?

2018  
1st KAKAO  
BLIND  
RECRUITMENT

① 차 코딩 테스트 문제 해설

2022 kakao  
blind recruitment

네이버 개발 직군 신입사원 공채

코딩 테스트 준비

우리는 신입 개발자들을 찾아낼 겁니다. 언제나 그랬듯이.

화려한 커리어보다는 러닝 커브에 주목합니다.

개발에 가슴 벅치고,

기초가 충실한 분이라면 자신 있게 도전하세요.

네이버의 미래를 개발하고 함께 성장할 분들을 기다립니다.

NAVER

이것이  
코딩 테스트다

with 파이썬



취업과 이직을 결정하는 알고리즘 인터뷰 완벽 가이드

이재명 지음



한글출판사

# 시간복잡도 왜 필요한가?

## 제한사항

- food\_times 는 각 음식을 모두 먹는데 필요한 시간이 음식의 번호 순서대로 들어있는 배열이다.
- k 는 방송이 중단된 시간을 나타낸다.
- 만약 더 섭취해야 할 음식이 없다면 **-1** 을 반환하면 된다.

## 정확성 테스트 제한 사항

- food\_times 의 길이는 **1** 이상 **2,000** 이하이다.
- food\_times 의 원소는 **1** 이상 **1,000** 이하의 자연수이다.
- k는 **1** 이상 **2,000,000** 이하의 자연수이다.

## 효율성 테스트 제한 사항

- food\_times 의 길이는 **1** 이상 **200,000** 이하이다.
- food\_times 의 원소는 **1** 이상 **100,000,000** 이하의 자연수이다.
- k는 **1** 이상 **2 x 10<sup>13</sup>** 이하의 자연수이다.

## 입출력 예

food_times	k	result
[3, 1, 2]	5	1

## 입출력 예 설명

### 입출력 예 #1

- 0~1초 동안에 1번 음식을 섭취한다. 남은 시간은 [2,1,2] 이다.
- 1~2초 동안 2번 음식을 섭취한다. 남은 시간은 [2,0,2] 이다.

<https://school.programmers.co.kr/learn/courses/30/lessons/42891>

# 시간복잡도 왜 필요한가?

---

BruteForcing

DP  
Greedy  
DFS, BFS  
BinarySearch

Segment Tree  
LCA  
Network Flow

# 시간복잡도 왜 필요한가?

---

1초    1억

# 시간복잡도 왜 필요한가?

---

## Big O Notation

```
for (int i = 0; i < N; i++)  
{  
    for (int j = 0; j < N; j++)  
    {  
    }  
}
```

$O(N^2)$

N = 100 10000

N = 1000 1000000



# 시간복잡도 왜 필요한가?

5 2751번

제출

맞힌 사람

숏코딩

재채점 결과

채점 현황

내 제출

🔗 난이도 기여

강의 ▾

질문 검색

수 정렬하기 2 

성공

☆

시간 제한	메모리 제한	제출	정답	맞힌 사람	정답 비율
2 초	256 MB	217280	62221	43221	30.441%

문제

N개의 수가 주어졌을 때, 이를 오름차순으로 정렬하는 프로그램을 작성하시오.

입력

첫째 줄에 수의 개수  $N(1 \leq N \leq 1,000,000)$ 이 주어진다. 둘째 줄부터 N개의 줄에는 수가 주어진다. 이 수는 절댓값이 1,000,000보다 작거나 같은 정수이다. 수는 중복되지 않는다.

출력

첫째 줄부터 N개의 줄에 오름차순으로 정렬한 결과를 한 줄에 하나씩 출력한다.

<https://www.acmicpc.net/problem/2751>

# 시간복잡도 왜 필요한가?

정렬 종류	시간 복잡도			공간 복잡도
	평균(Average)	최선(Best)	최악(Worst)	
선택 정렬	$O(n^2)$	$O(n^2)$	$O(n^2)$	$O(n^2)$
버블 정렬	$O(n^2)$	$O(n^2)$	$O(n^2)$	$O(n)$
삽입 정렬	$O(n^2)$	$O(n)$	$O(n^2)$	$O(n^2)$
합병 정렬	$O(n \times \log n)$	$O(n \times \log n)$	$O(n \times \log n)$	$O(n \times \log n)$
퀵 정렬	$O(n \times \log n)$	$O(n \times \log n)$	$O(n^2)$	$O(n \times \log n)$
힙 정렬	$O(n \times \log n)$	$O(n \times \log n)$	$O(n \times \log n)$	$O(n \times \log n)$
셸 정렬	$O(N^{1.25})$	$O(N^{1.25})$	$O(N^{1.25})$	$O(n)$
기수 정렬	$O(dn)$	$O(dn)$	$O(dn)$	

# 시간복잡도 왜 필요한가?

---

- for 문
- 재귀

대중적인 알고리즘의 시간복잡도는 알려져있음

+ 본인이 사용하는 내장함수의 시간복잡도는 알아야함

# 시간복잡도 왜 필요한가?

---

```
vector<vector<int>> adjacent; //인접 리스트로 구현된 그래프 들어감
vector<bool> visited; // 그래프의 정점만큼의 크기로 생성

void DFS(int now)
{
    visited[now] = true;

    for (int i = 0; i < adjacent[now].size(); i++)
    {
        int next = adjacent[now][i];
        if (!visited[next])
            DFS(next);
    }
}
```

$O(N+E)$

<https://currygamedev.tistory.com/10>

# 시간복잡도 왜 필요한가?

list

The Average Case assumes parameters generated uniformly at random.

Internally, a list is represented as an array; the largest costs come from growing or deleting somewhere near the beginning (because everything after the point has to be shifted one position to the right). For this reason, `collections.deque` is often a better choice for collections that need to be modified frequently at both ends.

Operation	Average Case	 Amortized Worst Case
Copy	$O(n)$	$O(n)$
Append[1]	$O(1)$	$O(1)$
Pop last	$O(1)$	$O(1)$
Pop intermediate[2]	$O(n)$	$O(n)$
Insert	$O(n)$	$O(n)$
Get Item	$O(1)$	$O(1)$
Set Item	$O(1)$	$O(1)$
Delete Item	$O(n)$	$O(n)$
Iteration	$O(n)$	$O(n)$
Get Slice	$O(k)$	$O(k)$
Del Slice	$O(n)$	$O(n)$
Set Slice	$O(k+n)$	$O(k+n)$
Extend[1]	$O(k)$	$O(k)$
 Sort	$O(n \log n)$	$O(n \log n)$
Multiply	$O(nk)$	$O(nk)$
x in s	$O(n)$	
min(s), max(s)	$O(n)$	
Get Length	$O(1)$	$O(1)$

<https://wiki.python.org/moin/TimeComplexity>

# Python vs C++

---

1. 코드의 실행 속도
2. 문자열 처리
3. 언어의 난이도/ 자료의 양

<https://blog.encrypted.gg/965>

<https://medium.com/@cookatrice/why-python-is-slow-looking-under-the-hood-7126baf936d7>

# Python vs C++

---

```
/* C code */  
  
int a = 1;  
  
int b = 2;  
  
int c = a + b;
```

## C 덧셈

1. <int> 1을 a에 할당
2. <int> 2을 b에 할당
3. binary\_add<int, int>(a,b) 호출
4. 결과를 c에 할당

a

1

b

2

# Python vs C++

---

```
# python code
```

```
a = 1
```

```
b = 2
```

```
c = a + b
```

1a. a->PyObject\_HEAD->typecode 정수 설정

1b. a->val = 1 설정

2. b에 2를 할당

2a. b->PyObject\_HEAD->typecode 정수 설정

2b. b->val = 2 설정

3. binary\_add(a,b) 호출

3a. a->PyObject\_HEAD 에서 typecode 찾기

3b. a는 정수형; 값 a->val

3c. b->PyObject\_HEAD 에서 typecode 찾기

3d. b는 정수형; 값 b->val

3e. binary\_add<int, int>(a->val, b->val) 호출

3f. 정수형 결과값 result

4. 파이썬 개체 c 생성

4a. c->PyObject\_HEAD->typecode 정수 설정

4b. c->val에 result 설정



# Python vs C++

---

1. 코드의 실행 속도
2. 문자열 처리
3. 언어의 난이도/ 자료의 양

<https://blog.encrypted.gg/965>

<https://medium.com/@cookatrice/why-python-is-slow-looking-under-the-hood-7126baf936d7>

# 이분탐색

---

19

1 4 5 13 18 19 21 30 31 35

$O(N)$

# 이분탐색



<https://www.acmicpc.net/blog/view/109>

$O(\log N)$

<https://velog.io/@ming/%EC%9D%B4%EB%B6%84%ED%83%90%EC%83%89Binary-Search>

# 이분 탐색

---

<https://www.acmicpc.net/problem/1072>

BruteForcing

BinarySearch

# 이분탐색

## 문제

김형택은 지금 몰래 Spider Solitaire(스파이더 카드놀이)를 하고 있다. 형택이는 이 게임을 이길 때도 있었지만, 질 때도 있었다. 누군가의 시선이 느껴진 형택이는 게임을 중단하고 코딩을 하기 시작했다. 의심을 피했다고 생각한 형택이는 다시 게임을 켜다. 그 때 형택이는 잠시 코딩을 하는 사이에 자신의 게임 실력이 눈에 띄게 향상된 것을 알았다.

이제 형택이는 앞으로의 모든 게임에서 지지 않는다. 하지만, 형택이는 게임 기록을 삭제 할 수 없기 때문에, 자신의 못하던 예전 기록이 현재 자신의 엄청난 실력을 증명하지 못한다고 생각했다.

게임 기록은 다음과 같이 생겼다.

- 게임 횟수 :  $X$
- 이긴 게임 :  $Y$  ( $Z\%$ )
- $Z$ 는 형택이의 승률이고, 소수점은 버린다. 예를 들어,  $X=53$ ,  $Y=47$ 이라면,  $Z=88$ 이다.

$X$ 와  $Y$ 가 주어졌을 때, 형택이가 게임을 최소 몇 번 더 해야  $Z$ 가 변하는지 구하는 프로그램을 작성하시오.

## 입력

각 줄에 정수  $X$ 와  $Y$ 가 주어진다.

## 출력

첫째 줄에 형택이가 게임을 최소 몇 판 더 해야하는지 출력한다. 만약  $Z$ 가 절대 변하지 않는다면 -1을 출력한다.

## 제한

- $1 \leq X \leq 1,000,000,000$
- $0 \leq Y \leq X$

Q & A