

컴퓨팅 사고와 SW 코딩

04

정렬과 탐색

경북대학교 배준현 교수
(joonion@knu.ac.kr)



04. 정렬과 탐색

- 정렬 문제: *sorting* problem
 - 무작위로 주어지는 항목을 키 값에 따라 순서대로 정렬하는 문제
 - 키 값: 하나일 수도 있지만, 여러 개일 수도 있음
 - 순서: 오름차순 또는 내림차순



04. 정렬과 탐색

- 버블 정렬

처음부터 차례대로 가서 가장 큰 애가 가장 뒤로 가게 하는 정렬 방식

순차적으로 탐색

- 선택 정렬

■ BOJ 2750: 수 정렬하기

- 문제:

- N개의 수가 주어졌을 때, 이를 오름차순으로 정렬하는 프로그램을 작성하시오.

- 입력:

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

- 출력:

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



04. 정렬과 탐색

예제 입력

5
5
2
3
4
1

예제 출력

1
2
3
4
5



04. 정렬과 탐색

```
N = int(input())
nums = []
for _ in range(N):
    nums.append(int(input()))
nums.sort()
for i in range(N):
    print(nums[i])
```

```
N = int(input())
nums = [int(input()) for _ in range(N)]
S = sorted(nums)
for i in range(N):
    print(nums[i])
```

```
N = int(input())
nums = [0] * N
for i in range(N):
    nums[i] = int(input())
nums.sort()
for i in range(N):
    print(nums[i])
```



04. 정렬과 탐색

- 오름차순이 아닌 **내림차순**으로 정렬하려면?

```
N = int(input())
nums = [int(input()) for _ in range(N)]
nums.sort(reverse = True)
for i in range(N):
    print(nums[i])
```



04. 정렬과 탐색

■ BOJ 2751: 수 정렬하기 2

- 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.



04. 정렬과 탐색

```
import sys
```

```
input = sys.stdin.readline
```

BOJ에서 input, output이 느려서 시간초과가 나지 않도록 해줌
input함수의 이름을 sys.stdin.readline으로 대체해줌

```
N = int(input())
```

```
nums = [int(input()) for _ in range(N)]
```

```
nums.sort()
```

```
for i in range(N):
```

```
    print(nums[i])
```




■ BOJ 11650: 좌표 정렬하기

- 문제:
 - 2차원 평면 위의 점 N 개가 주어진다.
 - 좌표를 x 좌표가 증가하는 순으로, x 좌표가 같으면 y 좌표가 증가하는 순서로 정렬한 다음 출력하는 프로그램을 작성하시오.
- 입력:
 - 첫째 줄에 점의 개수 N ($1 \leq N \leq 100,000$)이 주어진다.
 - 둘째 줄부터 N 개의 줄에는 i 번점의 위치 x_i 와 y_i 가 주어진다.
 - $(-100,000 \leq x_i, y_i \leq 100,000)$
 - 좌표는 항상 정수이고, 위치가 같은 두 점은 없다.
- 출력:
 - 첫째 줄부터 N 개의 줄에 점을 정렬한 결과를 출력한다.



04. 정렬과 탐색

예제 입력

5
3 4
1 1
1 -1
2 2
3 3

예제 출력

1 -1
1 1
2 2
3 3
3 4



04. 정렬과 탐색

```
import sys
input = sys.stdin.readline

N = int(input())
nums = [tuple(map(int, input().split())) for _ in range(N)]
nums.sort()
for i in range(len(nums)):
    print(" ".join(map(str, nums[i])))
```



04. 정렬과 탐색

- x좌표는 오름차순, y좌표는 내림차순으로 정렬하려면?

```
import sys
input = sys.stdin.readline

N = int(input())
nums = [list(map(int, input().split())) for _ in range(N)]
nums.sort(key = lambda x: (x[0], -x[1]))
for i in range(N):
    print(" ".join(map(str, nums[i])))
```

list는 mutable(바꿀 수 있음)
tuple은 immutable(바꿀 수 없음)



04. 정렬과 탐색

■ BOJ 1181: 단어 정렬

- 문제:

- 알파벳 소문자로 이루어진 N개의 단어가 들어오면 아래와 같은 조건에 따라 정렬하는 프로그램을 작성하시오.
 - 길이가 짧은 것부터
 - 길이가 같으면 사전 순으로

- 입력:

- 첫째 줄에 단어의 개수 N이 주어진다. ($1 \leq N \leq 20,000$)
- 둘째 줄부터 N개의 줄에 걸쳐 알파벳 소문자로 이루어진 단어가 한 줄에 하나씩 주어진다.
- 주어지는 문자열의 길이는 50을 넘지 않는다.

- 출력:

- 조건에 따라 정렬하여 단어들을 출력한다.
- 단, 같은 단어가 여러 번 입력된 경우에는 한 번씩만 출력한다.



04. 정렬과 탐색

예제 입력

13
but
i
wont
hesitate
no
more
no
more
it
cannot
wait
im
yours

예제 출력

i
im
it
no
but
more
wait
wont
yours
cannot
hesitate



04. 정렬과 탐색

```
import sys
input = sys.stdin.readline
n = int(input())
A = [input().strip() for _ in range(n)]
A = list(set(A))
A.sort(key = lambda x : (len(x), x))
print("\n".join(A))
```

04. 정렬과 탐색

■ 정렬 알고리즘: sorting algorithm

- 버블 정렬: Bubble Sort
- 선택 정렬: Selection Sort
- 삽입 정렬: Inserting Sort
- 합병 정렬: Merge Sort
- 빠른 정렬: Quick Sort
- 카운트 정렬: Counting Sort
- 기수 정렬: Radix Sort
- 버킷 정렬: Bucket Sort
- 그 외 다수: Shell Sort, Tim Sort, etc.

비교를 이용해서 정렬

$O(N^2)$

$O(n\log(n))$



04. 정렬과 탐색

- 탐색 문제: *searching* problem
 - 주어진 데이터로부터 **특정한 조건**을 만족하는 데이터를 찾는 문제
 - 또는, 주어진 데이터를 **최대한 빨리 찾을 수 있는 방식으로 저장**하는 문제
 - 선택 문제: *selection* problem
 - 무작위로 주어진 key의 집합에서 k-번째로 큰 key를 찾는 문제
 - $k = 1$: 최솟값, $k = n$: 최댓값, $k = \lfloor n/2 \rfloor$: 중앙값



04. 정렬과 탐색

■ BOJ 10818: 최소, 최대

- 문제:
 - N개의 정수가 주어진다.
 - 이때, **최솟값**과 **최댓값**을 구하는 프로그램을 작성하시오.
- 입력:
 - 첫째 줄에 정수의 개수 N ($1 \leq N \leq 1,000,000$)이 주어진다.
 - 둘째 줄에는 N개의 정수를 공백으로 구분해서 주어진다.
 - 모든 정수는 -1,000,000보다 크거나 같고, 1,000,000보다 작거나 같은 정수이다.
- 출력:
 - 첫째 줄에 주어진 정수 N개의 최솟값과 최댓값을 공백으로 구분해 출력한다.



04. 정렬과 탐색

예제 입력

5
20 10 35 30 7

예제 출력

7 35



04. 정렬과 탐색

```
N = int(input())
A = list(map(int, input().split()))
A.sort()
print(A[0], A[-1])
```

```
def solve(n, A):
    smallest, largest = 10**6, -10**6
    for i in range(n):
        smallest = min(smallest, A[i])
        largest = max(largest, A[i])
    return smallest, largest
```

```
N = int(input())
A = list(map(int, input().split()))
s, l = solve(N, A)
print(s, l)
```



04. 정렬과 탐색

■ BOJ 2501: 약수 구하기

• 문제:

- 두 개의 자연수 N 과 K 가 주어졌을 때,
 - N 의 약수들 중 K 번째로 작은 수를 출력하는 프로그램을 작성하시오.
- 어떤 자연수 p 와 q 가 있을 때,
 - 만일 p 를 q 로 나누었을 때 나머지가 0이면 q 는 p 의 약수이다.
 - 6을 예로 들면, 6의 약수는 1, 2, 3, 6, 총 네 개이다.

• 입력:

- 첫째 줄에 N 과 K 가 빈칸을 사이에 두고 주어진다.
- N 은 1 이상 10,000 이하이다. K 는 1 이상 N 이하이다.

• 출력:

- 첫째 줄에 N 의 약수들 중 K 번째로 작은 수를 출력한다.
- 만일 N 의 약수의 개수가 K 개보다 적어서 K 번째 약수가 존재하지 않을 경우에는 0을 출력



04. 정렬과 탐색

```
def solve(n, k):
    cnt = 0
    for d in range(1, n + 1):
        if n % d == 0:
            cnt += 1
        if cnt == k:
            return d
    return 0
```

```
N, K = map(int, input().split())
print(solve(N, K))
```

소수 판별 알고리즘

```
def is_prime(n):
    if n==1:
        return False
    else:
        for i in range(2, n):
            if n%i == 0:
                return False
        return False
```

```
def is_prime(n):
    if n==1:
        return False
    else:
        for i in range(2, sqrt(n)+1):
            if n%i == 0:
                return False
        return False
```

N부터 M까지 소수의 개수를 구하라
-> 동적계획법



04. 정렬과 탐색

- 모든 약수를 탐색하는 더 효율적인 방법은 없을까?

```
def divisor(n):  
    div = set()  
    for d in range(1, int(n ** 0.5) + 1):  
        if n % d == 0:  
            div.add(d)  
            div.add(n // d)  
    return sorted(list(div))
```

```
N, K = map(int, input().split())  
D = divisor(N)  
print(D[K - 1] if K <= len(D) else 0)
```





04. 정렬과 탐색

■ BOJ 5618: 공약수

- 문제:
 - 자연수 n 개가 주어진다.
 - 이 자연수의 공약수를 모두 구하는 프로그램을 작성하시오.
- 입력:
 - 첫째 줄에 n 이 주어진다. n 은 2 또는 3이다.
 - 둘째 줄에는 공약수를 구해야 하는 자연수 n 개가 주어진다.
 - 모든 자연수는 10^8 이하이다.
- 출력:
 - 입력으로 주어진 n 개 수의 공약수를 한 줄에 하나씩 증가하는 순서대로 출력한다.



04. 정렬과 탐색

예제 입력

2

75 125

3

110 22 88

3

66 11 3

예제 출력

1

5

25

1

2

11

22

1



04. 정렬과 탐색

```
def solve(n, A):  
    m = min(A)  
    for k in range(1, m + 1):  
        if A[0] % k == 0 and A[1] % k == 0:  
            if n == 2:  
                print(k)  
            elif n == 3 and A[2] % k == 0:  
                print(k)  
  
N = int(input())  
A = list(map(int, input().split()))  
solve(N, A)
```





04. 정렬과 탐색

- 모든 공약수를 탐색하는 더 효율적인 방법은 없을까?

```
def gcd(n, m):  
    if m == 0:  
        return n  
    else:  
        return gcd(m, n % m)  
  
def solve(n, A):  
    G = gcd(A[0], A[1])  
    if N == 3:  
        G = gcd(G, A[2])  
    for d in divisor(G):  
        print(d)  
  
N = int(input())  
A = list(map(int, input().split()))  
solve(N, A)
```



04. 정렬과 탐색

■ 더 풀어볼 문제:

- BOJ 1427: 소트인사이드
- BOJ 10814: 나이순 정렬
- BOJ 11651: 좌표 정렬하기 2
- BOJ 10448: 유레카 이론
- BOJ 2108: 통계학
- BOJ 1037: 약수
- BOJ 2609: 최대공약수와 최소공배수

Any Questions?

