Map, Set,Number theory

2020 Winter 초급

서강대학교 강효규

map && set

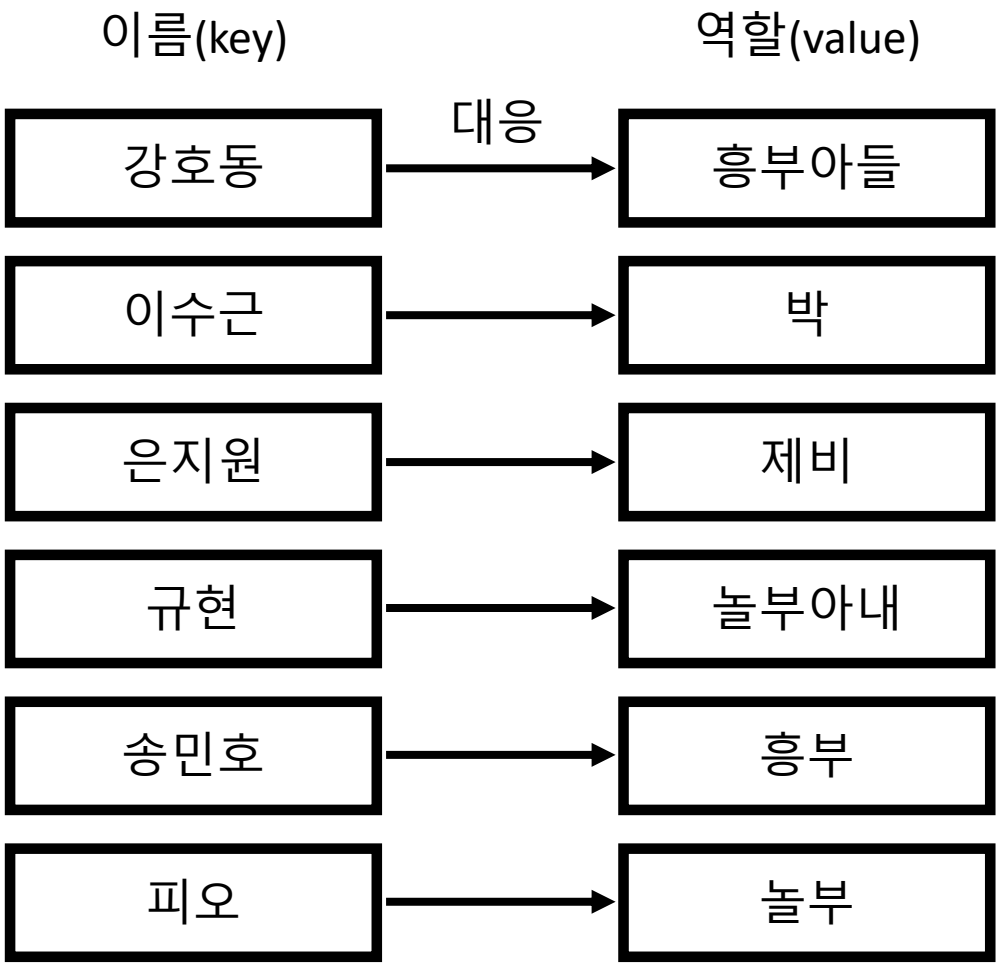
✓연관 컨테이너(associative containter) cf. 연속 컨테이너(sequence container)

✓key는 중복될 수 없음 cf. multimap, multiset

✓key를 기준으로 정렬된 상태

✓균형 이진탐색트리(Balanced Binary Search Tree) 구조

✓탐색, 삽입: $\boldsymbol{O(logN)}$



이름(key)을 이용해 이에 대응되는 역할(value)을 알 수 있다

map && set

✓map: 키(key) – 값(value) 구조

✓set: 키(key)값만 저장

즉, set은 중복되지 않는 값들이 오름차순으로 정렬되어 저장된 컨테이너

✓vector, list, array보다 빠르게 $\boldsymbol{O(logN)}$에 값 탐색 및 삽입 가능

map && set

반복자(iterator)

✓배열 및 컨테이너 내부 요소를 순회하는 객체

✓map, set은 선형 자료구조가 아니기 때문에 iterator를 통해 순회함

✓iterator를 역참조하여 값에 접근할 수 있음

✓map의 원소의 first에는 key, second에는 value가 저장되어 있음

```
1   map<int, int> mp;
2   for (map<int, int>::iterator iter = mp.begin(); iter != mp.end(); iter++)
3       cout << iter->first << ' ' << iter->second << '\n'; //key, value

4   for (auto iter = mp.begin(); iter != mp.end(); iter++)
5       cout << iter->first << ' ' << iter->second << '\n';
```

✓set도 iterator를 역참조하여 값에 접근 가능

```
6   set<int> st;
7   for (set<int>::iterator iter = st.begin(); iter != st.end(); iter++)
8       cout << *iter << '\n';

9   for (auto iter = st.begin(); iter != st.end(); iter++)
10      cout << *iter << '\n';
```

map

✓map 헤더 사용

✓key, value의 타입을 pair 형식으로 작성

```
1   map<int, int> mp;
```

원소 삽입

✓insert: 해당 key에 value가 대응된 원소 삽입

만약 key가 이미 존재할 경우 value 변경되지 않음

```
2   mp.insert({ 1, 11 });
3   mp.insert(make_pair(2, 22));
4   mp.insert({1, 111}); //value 변화 없음
```

원소 삽입

✓operator [ ]: key값을 갖는 원소 생성 (default: value = 0)

만약 key가 이미 존재할 경우 해당 원소의 value 변경

```
5   mp[1] = 111;
6   mp[2] = 222;
7   mp[3] = 333;
8   mp[4] += 444; //mp[4] = 444
```

원소 유무 확인

✓find : 해당 key를 갖는 원소의 iterator 반환

만약 해당 원소가 없을 경우 마지막 iterator 반환

```
 9   if (mp.find(1) != mp.end()) cout << "YES" << '\n';
10   else cout << "NO" << '\n';
```

원소 유무 확인

✓count: 해당 key를 갖는 원소 개수 반환

map은 중복된 key를 갖지 않으므로 원소가 있으면 1, 없으면 0을 반환함

```
11   if (mp.count(1)) cout << "YES" << '\n';
12   else cout << "NO" << '\n';
```

원소 값 변경

✓find

```
13  map<int, int>::iterator iter = mp.find(1);
14  if (iter != mp.end())
15      iter->second++;
```

✓operator [ ]

```
16  mp[1]++;
```

원소 삭제

✓erase: 해당 key를 갖는 원소가 있을 경우 이를 삭제함

```
13   mp.erase(1);
```

원소 개수

✓size: map의 원소 개수를 반환함

```
14    cout << mp.size() << '\n';
```

set

✓set 헤더 사용

```
1   #include <set>
2   set<int> st;
```

원소 삽입

✓insert: 해당 값을 갖는 원소 삽입

만약 해당 원소가 이미 있을 경우 set 구성에 변화 없음

```
3   st.insert(1);
4   st.insert(2);
5   st.insert(1); //변화  없음
```

원소 유무 확인

✓find : 해당 값을 갖는 원소의 iterator 반환

만약 해당 원소가 없을 경우 마지막 iterator 반환

```
6   if(st.find(1) != st.end()) cout << "YES" << '\n';
7   else cout << "NO" << '\n';
```

원소 유무 확인

✓count: 해당 값을 갖는 원소 개수 반환

set은 중복된 원소를 갖지 않으므로 원소가 있으면 1, 없으면 0을 반환함

```
8   if (st.count(1)) cout << "YES" << '\n';
9   else cout << "NO" << '\n';
```

원소 삭제

✓erase: 해당 값을 갖는 원소가 있을 경우 이를 삭제함

```
10   st.erase(1);
```

원소 개수

✓size: set의 원소 개수를 반환함

```
11    cout << st.size() << '\n';
```

#1302 베스트셀러

김형택은 탑문고의 직원이다. 김형택은 계산대에서 계산을 하는 직원이다. 김형택은 그날 근무가 끝난 후에, 오늘 판매한 책의 제목을 보면서 가장 많이 팔린 책의 제목을 칠판에 써놓는 일도 같이 하고 있다.

오늘 하루 동안 팔린 책의 제목이 입력으로 들어왔을 때, 가장 많이 팔린 책의 제목을 출력하는 프로그램을 작성하시오.

입력

첫째 줄에 오늘 하루 동안 팔린 책의 개수 N이 주어진다. 이 값은 1,000보다 작거나 같은 자연수이다. 둘째부터 N개의 줄에 책의 제목이 입력으로 들어온다. 책의 제목의 길이는 50보다 작거나 같고, 알파벳 소문자로만 이루어져 있다.

출력

첫째 줄에 가장 많이 팔린 책의 제목을 출력한다. 만약 가장 많이 팔린 책이 여러 개일 경우에는 사전 순으로 가장 앞서는 제목을 출력한다.

#1302 베스트셀러

```
14      cin >> n;
15      map<string, ll>mp;
16      while (n--) {
17          string s;
18          cin >> s;
19          if (mp.find(s) == mp.end())mp.insert({ s,1 });
20          else mp.find(s)->second++;
21          a = max(a, mp.find(s)->second);
22      }
23      map<string, ll>::iterator iter;
24      for(iter=mp.begin();iter!=mp.end();iter++)
25          if (iter->second == a) {
26              cout << iter->first;
27              return 0;
28          }
```

유클리드 호제법

✓두 자연수 $a, b\ (a > b)$, a를 b로 나눈 나머지 r

✓a와 b의 최대공약수는 b와 r의 최대공약수와 같음

✓b를 r로 나눈 나머지 r'를 구하고, 다시 r을 r'로 나눈 나머지를 구하는 과정을 반복

✓나머지가 0일 때 나누는 수가 a와 b의 최대공약수

유클리드 호제법

```
1  int gcd(int a, int b)
2  {
3      return b ? gcd(b, a%b) : a;
4  }
```

```
1  int gcd(int a, int b)
2  {
3      int c;
4      while(b)
5      {
6          c = a % b;
7          a = b;
8          b = c;
9      }
10     return a;
11 }
```

#2609 최대공약수와 최소공배수

두 개의 자연수를 입력받아 최대 공약수와 최소 공배수를 출력하는 프로그램을 작성하시오.

입력

첫째 줄에는 두 개의 자연수가 주어진다. 이 둘은 10,000이하의 자연수이며 사이에 한 칸의 공백이 주어진다.

출력

첫째 줄에는 입력으로 주어진 두 수의 최대공약수를, 둘째 줄에는 입력으로 주어진 두 수의 최소 공배수를 출력한다.

#2609 최대공약수와 최소공배수

최소공배수는 두수의 곱을 최대공약수로 나눈 것으로 구할 수 있다.

```
22      cin >> a >> b;
23      cout << gcd(a, b) << '\n';
24      cout << a * b / gcd(a, b) << '\n';
```

소수 판정법

✓n보다 작은 수로 모두 나눠보는 방식? -> O(n)

✓$\sqrt{n}$보다 작은 수로만 나눠봐도 된다! -> O($\sqrt{n}$)

✓만약 n이 $\sqrt{n}$ 이상의 자연수로 나누어진다면 그 몫은 2 이상 $\sqrt{n}$ 이하

✓따라서 2 이상 $\sqrt{n}$ 이하의 자연수로 나누어지지 않으면 $\sqrt{n}$ 초과 n 이하의 자연수에 대해서는 확인할 필요 없다

소수 판정법

```cpp
long long x; cin >> x;
bool isPrime = true;

for (long long i = 2; i * i <= x; i++) {
    if (x % i == 0) {
        isPrime = false;
        break;
    }
}
```

소수 판정법 – 에라토스테네스의 체

에라토스테네스의 체

✓각 수가 소수인지의 여부와 소수 목록 저장

✓O(1)에 소수 판정 가능

✓공간복잡도 포기

소수 판정법 – 에라토스테네스의 체

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

소수 판정법 – 에라토스테네스의 체

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

소수 판정법 – 에라토스테네스의 체

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

소수 판정법 – 에라토스테네스의 체

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

소수 판정법 – 에라토스테네스의 체

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

소수 판정법 – 에라토스테네스의 체

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

소수 판정법 – 에라토스테네스의 체

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

소수 판정법 – 에라토스테네스의 체

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

소수 판정법 – 에라토스테네스의 체

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

소수 판정법 – 에라토스테네스의 체

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

소수 판정법 – 에라토스테네스의 체

```
long long N; cin >> N;
vector<long long> primeNums;
vector<bool> isPrime(N + 1, true);

isPrime[0] = isPrime[1] = false;
for (long long i = 2; i <= N; i++) {
    if (isPrime[i]) {
        primeNums.push_back(i);
        for (long long j = i * i; j <= N; j += i) {
            isPrime[j] = false;
        }
    }
}
```

소인수 분해

✓나누어 떨어지는 수가 나올 때 마다 계속해서 나눠준다.

✓$\sqrt{N}$ 보다 큰 숫자들에 대해서 나누어 떨어지는지 판단 할 필요가 없다.

```
2  void func(ll x) {
3      for (ll i = 2; i * i <= x; i++) {
4          if (x % i)continue;
5          while (x % i==0) {
6              cout << i << '\n';
7              x /= i;
8          }
9      }
10     if(x!=1)
11     cout << x;
12 }
```

#추천문제

필수문제

14490	3 백대열
20302	5 민트 초코
16563	5 어려운 소인수분해
4358	4 생태학
2002	4 추월

연습문제

7785	5 회사에 있는 사람
1978	4 소수 찾기
1620	4 나는야 포켓몬 마스터 이다솜
6588	1 골드바흐의 추측
19583	1 싸이버개강총회
20003	1 거스름돈이 싫어요
1963	4 소수 경로
15711	4 환상의 짝꿍
19566	2 수열의 구간 평균
1016	1 제곱 ㄴㄴ 수
2882	3 소수 사이클