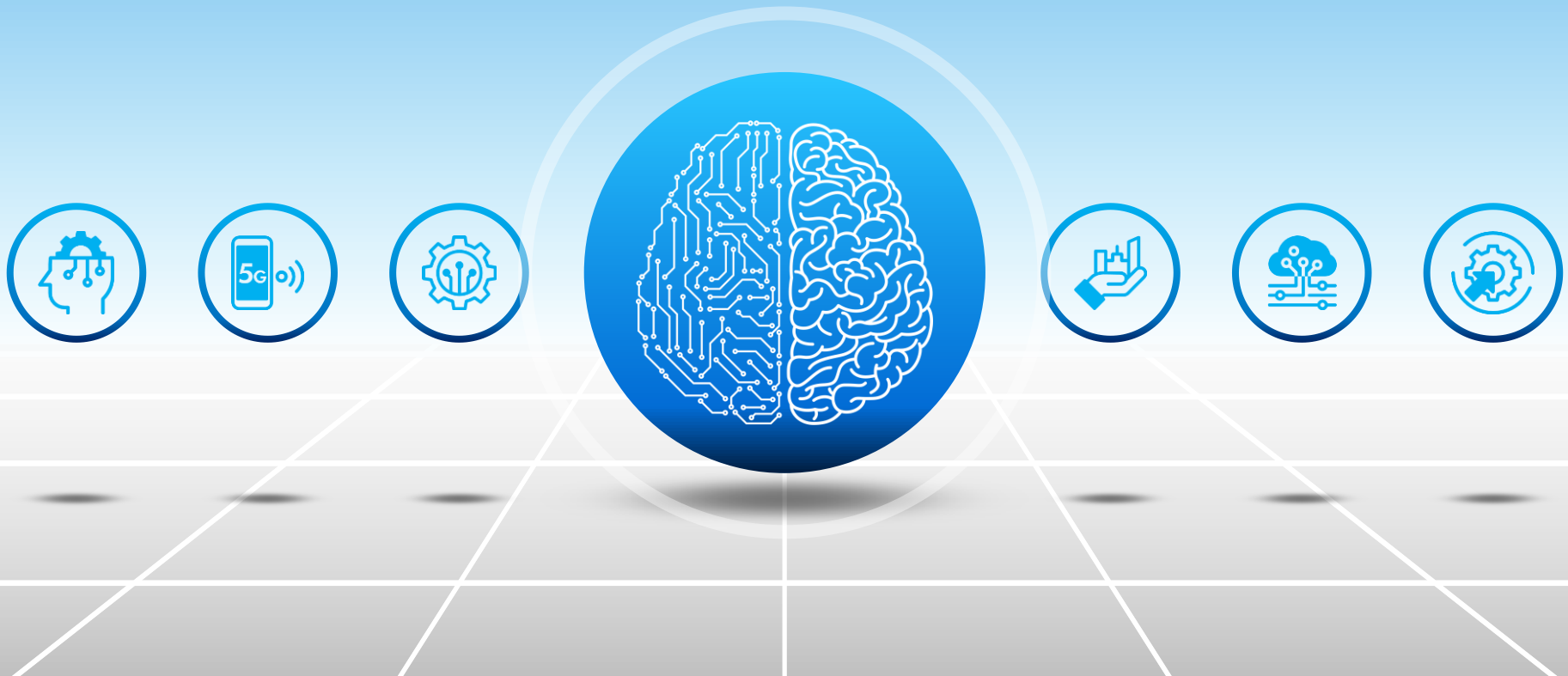


CIS3034 문제해결프로젝트

맵을 활용한 문제해결



목 차

CONTENTS



I 맵

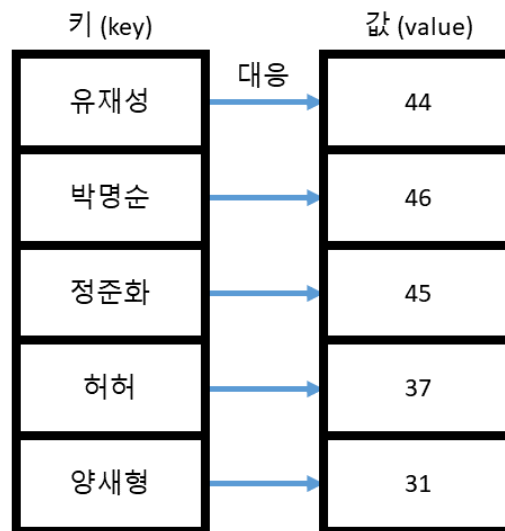
II 맵 예제 풀이

맵의 이해



• 맵(Map)

- 인덱스로 int가 아닌 다른 자료형을 사용할 수 있는 자료구조
- 내부적인 구조는 각 노드가 key와 value의 쌍으로 이루어진 트리
로 되어 있음.
- Key는 중복을 허용하지 않고, value는 중복을 허용함.
- Key 중복을 허용하기 위해서는 MultiMap을 사용해야 함.
- 특히 검색, 삽입, 삭제 등의 속도가 $O(\log_2 N)$ 으로 빠름.



이름(key)을 바탕으로 나이(value)를 얻을 수 있다.

• 맵의 특징

- 일반적인 성질은 Set과 유사함.
- Set과는 다르게 key와 value를 동시에 관리함.
- 인덱스(key)를 중복해서 추가할 경우, 기존 데이터가 새로운 데이터로 덮어 씌여짐.
- 반복자(iterator)와 배열 첨자를 이용하여 접근할 수 있음.
- key를 기준으로 오름차순으로 정렬하므로 검색 속도가 매우 빠름

• 맵의 용도

- 연관 있는 두 값을 함께 묶어서 관리하되, 검색을 빠르게 하고 싶은 경우
 - 이름과 몸무게, 이름과 주민등록번호 등
 - 지역과 지역번호, 지역과 온도 등

• 맵의 사용 사례

- 헤더파일 추가

```
#include <map>
```

- 표준 네임스페이스 사용하면 편리

```
using namespace std;
```

- 맵 변수 선언

```
map<key datatype, value datatype> 변수명;
```

```
map<string, int> m;
```

- 원소 접근

- 반복자(iterator)를 이용하는 방식
- 인덱스(key)를 이용하는 방식

• 맵의 사용 사례

멤버 함수	기능
m.size()	m의 노드 개수를 리턴
m.empty()	m의 사이즈가 0인지 아닌지를 확인
m.begin()	m의 첫 번째 원소를 가리키는 iterator 리턴
m.end()	m의 마지막 원소를 가리키는 iterator 리턴 (정확히는 마지막 뒤 공백 부분을 가리킴)
m[k] = v	m에 key가 k이고, value가 v인 노드 추가
m.insert(make_pair(k,v))	
m.erase(k)	m에서 key가 k인 노드 삭제
m.find(k)	m에서 key가 k인 노드를 찾아, 해당 노드를 가리키는 iterator 리턴 (key가 k인 노드가 존재하지 않는 경우, m의 마지막 원소를 가리키는 iterator 리턴)
m.count(k)	m에서 key가 k인 노드의 개수를 리턴

- Insert, erase 함수의 경우, 파라미터로 값 자체가 아닌 반복자를 넘겨주는 방식을 사용할 수도 있음.

ex1) vector v에 있는 모든 값 추가 -> m.insert(v.begin(), v.end())

ex2) map의 첫 번째 원소 삭제 -> m.erase(m.begin())

• 맵의 사용 사례

```

1 map<char,int> m;
2 map<char,int>::iterator it;
3
4 m['B'] = 2; //m : (B,2)
5 m.insert(make_pair('A',1)); //m : (A,1) (B,2)
6 m['C'] = 3; //m : (A,1) (B,2) (C,3)
7
8 m.erase('A'); //m : (B,2) (C,3)
9
10
11 //m전체를 순회하며 key와 value 출력
12 for(it = m.begin(); it != m.end(); it++)
13     cout << it->first << ' ' << it->second << '\n';
14
15 if(m.find('B') != m.end())
16     cout << "key값이 B인 노드가 존재합니다." << '\n';
17 else
18     cout << "key값이 B인 노드가 존재하지 않습니다." << '\n';

```

멤버 함수

m.size()
m.empty()
m.begin()
m.end()
m[k] = v
m.insert(make_pair(k,v))
m.erase(k)
m.find(k)
m.count(k)

```

B 2
C 3
key값이 B인 노드가 존재합니다.

```


• 맵의 사용 사례

```

map<string,int> m; // Map 생성. Key=String, Value=int형
// Map에 Data를 삽입하는 방법 1
m.insert(pair<string,int>(string("서울"),1000));
m.insert(pair<string,int>("부산",500));
m["대전"]=400; // Map에 Data를 삽입하는 방법 2
m["대구"]=300;
m["광주"]=200;
m["인천"]=100;
m["독도"]=1;

m.erase(m.begin());
m.erase("인천");

map<string,int>::iterator it;
for (it=m.begin();it!=m.end();it++) {
    cout << it->first << ":" << it->second << "만명" << endl;
}
    
```

대구 :300만명
 대전 :400만명
 독도 :1만명
 부산 :500만명
 서울 :1000만명

• 맵의 사용 사례

```
#include <map>
#include <string>
#include <stdio.h>

using namespace std;
```

```
map<int, string> iMap;

iMap[5] = "5번요소의 데이터"; //요소에 대한 데이터를 임의 순서로 추가
iMap[3] = "3번요소의 데이터";
iMap[9] = "9번요소의 데이터";

map<int, string>::iterator itMap;

printf("iMap\n");
for(itMap = iMap.begin(); itMap != iMap.end(); itMap++)
    printf("%d : %s\n", itMap->first, itMap->second.c_str()); //정렬된 상태
printf("\n");

iMap[9] = "안녕하세요!"; //덮어 씌우기

printf("iMap\n");
for(itMap = iMap.begin(); itMap != iMap.end(); itMap++)
    printf("%d : %s\n", itMap->first, itMap->second.c_str());
printf("\n");

printf("iMap[5] = %s\n", iMap[5].c_str()); //인덱스를 통한 접근
```

```
map<char*, string> sMap; //char*과 string 둘다 문자열이지만
                        //char*는 정렬되지 않는다.
```

```
sMap["미국"] = "United State of America";
sMap["중국"] = "China";
sMap["일본"] = "Japan";
sMap["한국"] = "Korea, Republic";
```

```
map<char*, string>::iterator itStrMap;
```

```
printf("sMap\n");
for(itStrMap = sMap.begin(); itStrMap != sMap.end(); itStrMap++)
    printf("%s : %s\n", itStrMap->first, itStrMap->second.c_str());
printf("\n"); //마지막으로 넣은 한국이 가장 먼저 출력됨
```

```
sMap.erase("일본"); //일본을 삭제
```

```
printf("sMap\n");
for(itStrMap = sMap.begin(); itStrMap != sMap.end(); itStrMap++)
    printf("%s : %s\n", itStrMap->first, itStrMap->second.c_str());
printf("\n");
```

```
iMap
3 : 3번요소의 데이터
5 : 5번요소의 데이터
9 : 9번요소의 데이터
iMap
3 : 3번요소의 데이터
5 : 5번요소의 데이터
9 : 안녕하세요!
iMap[5] = 5번요소의 데이터
sMap
한국 : Korea, Republic
일본 : Japan
중국 : China
미국 : United State of America
sMap
한국 : Korea, Republic
중국 : China
미국 : United State of America
```

맵 예제 풀이



문제6J. 빈도수세기

N 개의 정수가 차례로 입력으로 주어진다.

각 입력에 대하여 아래와 같은 형식으로 입력 결과를 출력한다.

- C F 형식으로 세 정수를 출력한다.
 - C 는 현재까지 입력된 숫자의 종류의 수이다.
 - F 는 이 번에 입력된 숫자가 등장한 횟수이다.

예를 들어서 차례대로 $\{5, 3, 5, 3, 2\}$ 가 입력으로 주어진 경우 출력 결과는 차례로 $\{1\ 1, 2\ 1, 2\ 2, 2\ 2, 3\ 1\}$ 가 된다.

입력 형식

첫 줄에는 차례로 입력으로 주어질 정수들의 총 개수를 나타내는 20만이하의 자연수 N 이 주어진다. 이후 총 N 줄에 걸쳐서 한 줄에 하나 씩 입력이 주어진다.

- 고려해야 할 순서대로 입력이 주어진다.
- 모든 숫자는 32비트 정수형임이 보장된다.

출력 형식

각 N 개의 숫자에 대한 처리 결과를 한 줄마다 출력한다.

- 현재까지 입력으로 주어진 숫자의 종류와, 해당 숫자가 등장한 횟수를 공백으로 구분하여 한 줄에 출력한다.

문제 6J. 빈도수세기

입/출력 예시



: 공백



: 줄바꿈



: 탭

예시 1

입력

5

3

5

3

5

2



출력

1.1

2.1

2.2

2.2

3.1



문제6J. 빈도수세기

```
#include <stdio.h>
#include <map>

using namespace std;

int main() {

    int N;
    scanf("%d", &N);

    // 각 <정수, 빈도수> 형태로 key-value를 저장할 Map 자료구조
    // frequencyMap := 이전에 입력된 정수들의 빈도수를 저장하고 있다.
    map<int, int> frequencyMap;

    for (int i = 0; i < N; i++) {
        int X;
        scanf("%d", &X);
        frequencyMap[X]++;
        printf("%d %d\n", frequencyMap.size(), frequencyMap[X]);
    }

    return 0;
}
```

문제6K. 시장 추천하기

이상의 나라의 한 도시에서는 독특한 방법으로 시장을 선발한다. 이 도시에서는 따로 후보자를 선발하거나 번호를 부여하지 않으며, 해당 시의 시민 모두가 잠정적 후보가 된다. 모든 투표권자는 시장으로 추천하는 사람의 이름을 투표용지에 적어서 투표에 참가한다. 개표시에는 가장 많은 표에서 추천 받은 사람이 시장직을 권유받게 된다.

하지만 투표용지를 개표하는 사람의 입장에서는 후보 번호가 아니라 불특정 다수의 이름으로 개표를 해야하므로 상당히 고통스러운 작업이 된다. N 개의 투표용지와 각 투표용지에 적힌 후보자의 이름이 주어졌을 때 가장 많은 표를 얻은 후보자의 이름을 출력하는 프로그램을 작성하시오. 단, 모든 후보자는 서로 다른 이름을 가지고 있으며 같은 표수를 얻은 후보자가 여러명 존재 할 수 있다.

입력 형식

첫 줄에는 투표용지의 수를 나타내는 10만이하의 자연수 N 이 주어진다.

이후 총 N 줄에 걸쳐서 한 줄에 하나씩 해당 투표용지에 적힌 후보자의 이름이 주어진다.

- 후보자의 이름은 1~10글자의 알파벳 대문자로 구성된 문자열이다.
- 두 후보가 같은 이름을 가지는 경우는 없다.

출력 형식

첫 줄에는 가장 많은 표를 획득한 후보자의 표수를 출력한다.

두 번째 줄에는 해당 표수만큼 표를 획득한 후보들의 이름을 공백으로 구분하여 사전순-오름차순으로 출력한다.

입/출력 예시

 : 공백  : 줄바꿈  : 탭

예시 1

입력

```
5
DO
DO
HI
HI
BYE
```

출력

```
2
DO HI
```

문제6K. 시장 추천하기

```
#include <iostream>
#include <string>
#include <map>

using namespace std;

int main() {
    int N;
    cin >> N;

    // 각 후보 이름과 득표 수를 저장하는 key-value Map
    map<string, int> frequencyMap;

    int maxFrequency = 0; // 가장 많은 득표수

    // 각 이름이 하나 추가될 때마다 현재까지의 최대 득표 값 갱신, 최대 득표 후보 리스트를 갱신해간다.
    for (int i = 0; i < N; i++) {
        string st;
        cin >> st;
        frequencyMap[st]++;
        int k = frequencyMap[st];
        if (k > maxFrequency) {
            maxFrequency = k;
        }
    }

    // 최대 득표 후보 수를 출력한다.
    cout << maxFrequency << endl;

    // 최대 득표를한 동점 후보들 이름을 사전순으로 출력한다.
    map<string, int>::iterator it;
    for(it=frequencyMap.begin(); it!=frequencyMap.end(); it++)
    {
        if(it->second == maxFrequency)
            printf("%s ", it->first.c_str());
    }
}
```