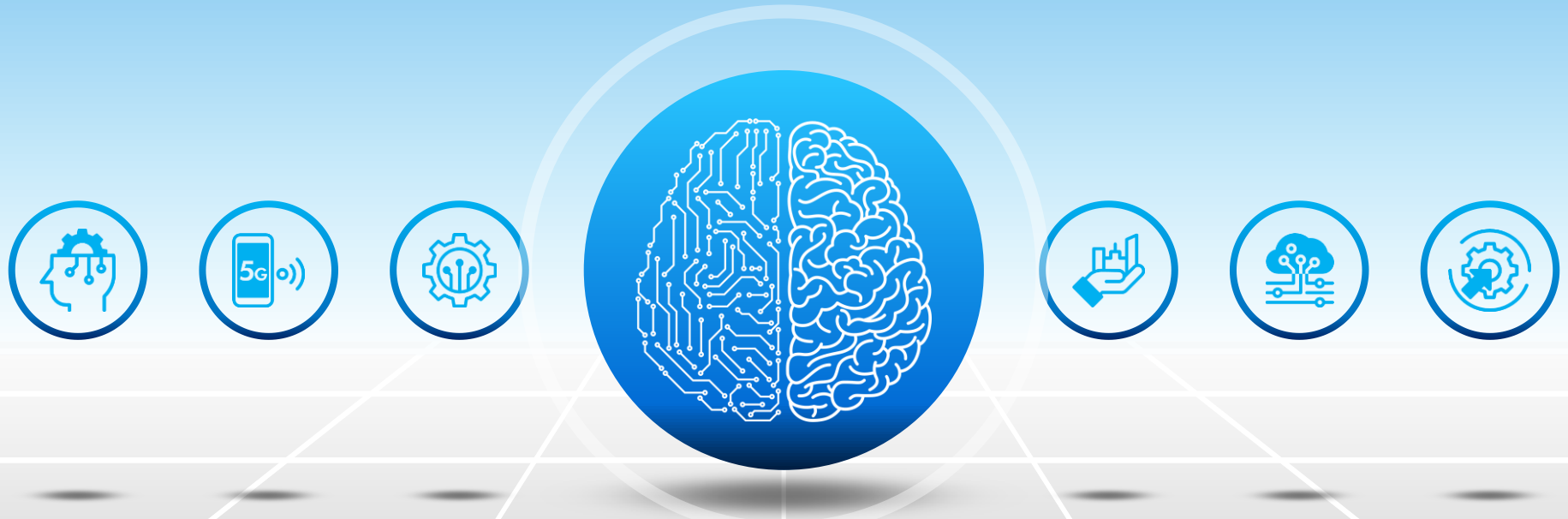
CIS3034 문제해결프로젝트

도형을 활용한 문제해결



목 차
CONTENTS

내장 기능 활용하기

정렬 함수

- **정렬은 대표적인 전처리 기법 중 하나이다.**
  - 데이터가 많을수록 연산량이 증가하지만, 정렬된 배열은 다양한 특징을 활용해 연산량을 최적화할 수 있다.
  - 이전에 실습한 정렬알고리즘(선택정렬, 버블정렬)은 $O(N^2)$의 시간복잡도를 가짐.
  - 대부분의 언어는 $O(N\log_2 N)$의 시간복잡도로 데이터를 정렬할 수 있는 내장 함수를 제공함.

C++	Java	Python
<algorithm>헤더의 sort()	Arrays.sort() 혹은 Collections.sort()	기본 리스트의 메소드 l.sort()
int arr[]={6,3,2}; sort(arr, arr+3);	int[] arr = new int[]{6,3,2}; Arrays.sort(arr);	l = [6, 3, 2] l.sort()

가변길이 배열

- **C++ 와 Java는 길이를 가변할 수 있는 vector와 ArrayList를 제공한다.**
  - 인덱스는 항상 0부터 연속적으로 부여된다.
  - 원소를 끝에 추가하는 것은 O(1)의 시간복잡도를 가진다.
  - 단, 중간의 데이터를 삭제하는 것은 O(N)의 시간복잡도를 가진다.
  - 데이터의 수를 미리 예측하기 어려울 때 효과적이다.

- **Python은 기본 list 구조가 가변길이 배열이며, .append() 내장함수로 데이터 추가가 가능하다.**

가변길이 배열

- C++ 와 Java는 길이를 가변할 수 있는 vecto와 ArrayList를 제공한다.

C++	Java
`vector<int> v;`	`ArrayList<Integer> a = new ArrayList<>();`

`v.push_back(5);`
`v.push_back(3);`
`v.push_back(7);`
	`a.add(5);`
	`a.add(3);`
	`a.add(7);`

`int value = v[2];`	`int value = a.get(2);`
`int len = v.size();`	`int len = a.size();`
`sort(v.begin(), v.end());`	`Collections.sort(a);`

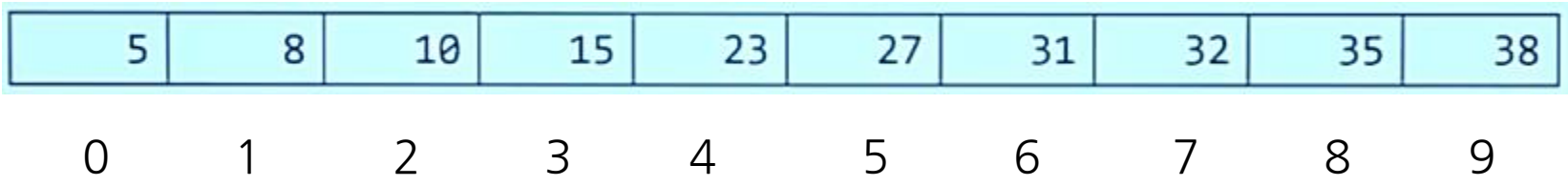
바이너리 서치

- **배열에서 바이너리 서치란?**
  - **정렬**된 데이터들 중 원하는 값을 빠르게 찾는 알고리즘을 말한다.
  - 찾고자 하는 값의 존재 여부나 위치를 $O(\log_2 N)$의 시간복잡도로 탐색할 수 있다.

C++	Java
<algorithm>헤더의 binary_search()	Arrays.binarySearch() 혹은 Collections.binarySearch()

바이너리 서치 구현하기

- 정렬된 데이터에서는 왜 빠르게 탐색을 할 수 있을까?



예제 풀이

문제31-세 카드

지수가 살고 있는 이상한 나라에서는 독특한 복권제도가 존재한다. 이상한 나라에서는 매 주 당첨될 자연수 번호를 정해두며, 복권을 구매한 사람은 그 자리에서 수 많은 카드들 중 하나를 뽑을 수 있는 기회가 세 번 주어진다. 즉, 똑같은 카드를 세 번 뽑을 수 도 있다. 이렇게 세 번에 걸쳐 뽑은 카드들에 적혀있는 세 자연수를 더하여 당첨 번호로 지정된 자연수와 일치한다면 그 사람은 당첨되는 것이다.

복권 담당자인 미주는 이번 주에 복권에 사용 될 당첨 번호들을 정하려고 한다. 하지만 매 번 실제로 그 당첨번호가 세 카드에 적힌 숫자들의 합으로 만들어 질 수 있는지 (즉, 실제로 당첨될 수 있는 번호인지) 검사하는 과정이 너무 번거로워 고민을 하고 있다. 미주를 도와서 주어진 카드를 조합해 당첨 번호들을 만들 수 있는지 판단하는 프로그램을 작성해주자.

문제31-세 카드

입력 형식

첫 줄에는 사용할 카드의 수 N과 당첨 번호의 숫자 M이 공백으로 구분되어 주어진다. N은 1이상 1,000이하의 자연수이며 M은 1이상 100이하의 자연수이다.

두 번째 줄에는 N개의 카드에 적힌 숫자들이 공백으로 구분되어 1이상 1억 이하의 자연수로 주어진다.

세 번째 줄에는 M개의 이번 주에 사용 될 당첨번호들이 공백으로 구분되어 주어진다. 당첨번호들은 모두 서로다른 1이상 3억 이하의 자연수이다.

출력 형식

M개의 당첨번호 들 중 실제로 세 카드에 적힌 숫자의 합으로 표현될 수 있는 당첨번호들을 모두 출력한다.

- 실제로 만들 수 있는 당첨번호가 여러개라면, 오름차순으로 정렬하고 각 숫자는 공백으로 구분하여 한 줄에 출력한다.
- 실제로 만들 수 있는 당첨번호가 존재하지 않는다면 NO를 출력한다.

문제31-세 카드

예시 1

입력

```
3 5
1 2 3
1 2 3 4 5
```

출력

```
3 4 5
```

예시 2

입력

```
10 5
1 2 3 4 5 6 7 8 9 10
1 2 20 30 40
```

출력

```
20 30
```

문제31-세 카드

```cpp
#include <stdio.h>
#include <vector>
#include <algorithm>

using namespace std;

/**
* 중복을 포함해 두 카드의 합으로 만들 수 있는 당첨번호의 수를 계산하는 함수
* @param n ····· 카드의 수
* @param m ····· 검사하려는 당첨번호의 수
* @param cards ··· 각 카드에 적힌 숫자들
* @param target ·· 검사하려는 각 당첨번호 리스트
* @return
*/
vector<int> getPossibleTargets(int n, int m, int * cards, int* targets) {
  vector<int> possibleTargets; // 만들 수 있는 당첨 번호들

  return possibleTargets;
}
```

```cpp
int main() {
  int n;  // 카드의 수
  int m;  // 검사 할 후보 당첨번호의 수
  scanf("%d %d", &n, &m);

  int* cards = new int[n];
  int* targets = new int[m];

  // 각 카드 정보를 입력받는다
  for (int i = 0; i < n; i++){
    scanf("%d", &cards[i]);
  }

  // 각 후보 당첨번호를 입력받는다
  for (int i = 0; i < m; i++){
    scanf("%d", &targets[i]);
  }

  vector<int> answers = getPossibleTargets(n, m, cards, targets);

  if (answers.size() == 0)
  {   // 가능한 당첨번호가 없다면 NO를 출력한다
    printf("NO");
  }
  else
  { //가능한 당첨번호가 존재한다면 그 목록을 출력한다.
    for (int i = 0; i < answers.size(); i++)
    {
      if (i > 0)
      {
        printf(" ");
      }
      printf("%d", answers[i]);
    }
  }

  delete[] cards;
  delete[] targets;

  return 0;
}
```

문제5H-두 직사각형

2차원 평면상에서 좌표축에 수직하거나 수평한 네 선분으로 구성된 직 사각형이 두 개 주어진다. 이 두 직사각형이 교차하는 영역의 넓이를 계산하는 프로그램을 작성하시오.

입력 형식

첫 줄에는 테스트케이스의 수를 나타내는 100이하의 자연수 T가 주어진다. 이후 총 T줄에 걸쳐서 각 테스트케이스에 대한 입력이 주어진다.

각 테스트케이스는 한 줄에 네 점 A, B, P, Q에 대한 좌표를 `Ax Ay Bx By Px Py Qx Qy`형식으로 주어진다. 모든 좌표는 절대값이 1만 이하인 정수이다.

- A와 B는 첫 번째 직사각형을 이루는 서로 대각선으로 마주보는 두 점의 좌표이다.
- P와 Q는 두 번째 직사각형을 이루는 서로 대각선으로 마주보는 두 점의 좌표이다.

출력 형식

각 테스트케이스에 대하여 한 줄에 두 직사각형이 교차하는 영역의 넓이를 정수로 출력한다.

문제5H-두 직사각형

입/출력 예시

␣ : 공백 ↲ : 줄바꿈 ↹ : 탭

예시 1

입력

```
1↲
1␣1␣3␣3␣4␣2␣2␣4
```

출력

```
1
```

문제5H-두 직사각형

```cpp
#include <iostream>
#include <cstdio>

using namespace std;

int get_area(int la, int ra, int ta, int ba,
             int lb, int rb, int tb, int bb)
{


  return 0;
}
```

```cpp
int main() {

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

  for(int i=0; i<t; i++)
    test_case();

  return 0;
}
```

```cpp
void test_case()
{
  int ax, ay, bx, by;
  int px, py, qx, qy;
  scanf("%d %d %d %d", &ax, &ay, &bx, &by);
  scanf("%d %d %d %d", &px, &py, &qx, &qy);

  int la, ra, ba, ta;//직사각형 a
  la = min(ax, bx);
  ra = max(ax, bx);
  ta = max(ay, by);
  ba = min(ay, by);

  int lb, rb, bb, tb;//직사각형 b
  lb = min(px, qx);
  rb = max(px, qx);
  tb = max(py, qy);
  bb = min(py, qy);

  int answer = get_area(la, ra, ta, ba,
                        lb, rb, tb, bb);

  printf("%d\n", answer);
}
```

실습 과제

- **7주차 2개 문제해결 프로그램 작성**
  - 세 카드
  - 두 직사각형
- **실습과제 게시판에 업로드할 것**
  - 2개의 cpp 파일
  - 파일명은 세 카드.cpp, 두 직사각형.cpp로 할 것
  - Cpp 파일에 코드를 설명할 수 있는 본인만의 주석을 작성할 것
- **8주차 프로젝트 과제 제시 예정**
  - 시험 및 강의 미실시