21/01/13

1. 체육복

점심시간에 도둑이 들어, 일부 학생이 체육복을 도난당했습니다. 다행히 여벌 체육복이 있는 학생이 이들에게 체육복을 빌려주려 합니다. 학생들의 번호는 체격 순으로 매겨져 있어, 바로 앞번호의 학생이나 바로 뒷번호의 학생에게만 체육복을 빌려줄 수 있습니다. 예를 들어, 4번 학생은 3번 학생이나 5번 학생에게만 체육복을 빌려줄 수 있습니다. 체육복이 없으면 수업을 들을 수 없기 때문에 체육복을 적절히 빌려 최대한 많은 학생이 체육수업을 들어야 합니다.

전체 학생의 수 n, 체육복을 도난당한 학생들의 번호가 담긴 배열 lost, 여벌의 체육복을 가져온 학생들의 번호가 담긴 배열 reserve가 매개변수로 주어질 때, 체육수업을 들을 수 있는 학생의 최댓값을 return 하도록 solution 함수를 작성해주세요.

제한사항

* 전체 학생의 수는 2명 이상 30명 이하입니다.
* 체육복을 도난당한 학생의 수는 1명 이상 n명 이하이고 중복되는 번호는 없습니다.
* 여벌의 체육복을 가져온 학생의 수는 1명 이상 n명 이하이고 중복되는 번호는 없습니다.
* 여벌 체육복이 있는 학생만 다른 학생에게 체육복을 빌려줄 수 있습니다.
* 여벌 체육복을 가져온 학생이 체육복을 도난당했을 수 있습니다. 이때 이 학생은 체육복을 하나만 도난당했다고 가정하며, 남은 체육복이 하나이기에 다른 학생에게는 체육복을 빌려줄 수 없습니다.

입출력 예

| **n** | **lost** | **reserve** | **return** |
| --- | --- | --- | --- |
| 5 | [2, 4] | [1, 3, 5] | 5 |
| 5 | [2, 4] | [3] | 4 |
| 3 | [3] | [1] | 2 |

입출력 예 설명

예제 #1  
1번 학생이 2번 학생에게 체육복을 빌려주고, 3번 학생이나 5번 학생이 4번 학생에게 체육복을 빌려주면 학생 5명이 체육수업을 들을 수 있습니다.

예제 #2  
3번 학생이 2번 학생이나 4번 학생에게 체육복을 빌려주면 학생 4명이 체육수업을 들을 수 있습니다.

* 내 답

#include <string>

#include <vector>

#include <set>

#include <algorithm>

using namespace std;

int solution(int n, vector<int> lost, vector<int> reserve) {

int answer = n;

set<int> reserve\_set;

for(int i=0;i<reserve.size();i++)

reserve\_set.insert(reserve[i]);

for(int i=0;i<lost.size();i++) {

if(reserve\_set.count(lost[i])) {

reserve\_set.erase(lost[i]);

continue;

}

if(reserve\_set.count(lost[i]-1)) {

reserve\_set.erase(lost[i]-1);

continue;

}

else if(reserve\_set.count(lost[i]+1) && find(lost.begin(),lost.end(),lost[i]+1)==lost.end()) {

reserve\_set.erase(lost[i]+1);

continue;

}

else

answer--;

}

return answer;

내 코드는 O(nlogn)의 시간복잡도가 나온다.

* 남의 답

#include <string>

#include <vector>

using namespace std;

int student[35];

int solution(int n, vector<int> lost, vector<int> reserve) {

int answer = 0;

for(int i : reserve) student[i] += 1;

for(int i : lost) student[i] += -1;

for(int i = 1; i <= n; i++) {

if(student[i] == -1) {

if(student[i-1] == 1)

student[i-1] = student[i] = 0;

else if(student[i+1] == 1)

student[i] = student[i+1] = 0;

}

}

for(int i = 1; i <=n; i++)

if(student[i] != -1) answer++;

return answer;

}

이 알고리즘은 O(n)의 시간복잡도이다. 굳이 set을 활용하지 않아도 이런식으로도 해결이 가능한 것 같은데 너무 STL 라이브러리에 의존하지 않도록 해야할 것 같다.

1. K번째 수

**문제 설명**

배열 array의 i번째 숫자부터 j번째 숫자까지 자르고 정렬했을 때, k번째에 있는 수를 구하려 합니다.

예를 들어 array가 [1, 5, 2, 6, 3, 7, 4], i = 2, j = 5, k = 3이라면

1. array의 2번째부터 5번째까지 자르면 [5, 2, 6, 3]입니다.
2. 1에서 나온 배열을 정렬하면 [2, 3, 5, 6]입니다.
3. 2에서 나온 배열의 3번째 숫자는 5입니다.

배열 array, [i, j, k]를 원소로 가진 2차원 배열 commands가 매개변수로 주어질 때, commands의 모든 원소에 대해 앞서 설명한 연산을 적용했을 때 나온 결과를 배열에 담아 return 하도록 solution 함수를 작성해주세요.

제한사항

* array의 길이는 1 이상 100 이하입니다.
* array의 각 원소는 1 이상 100 이하입니다.
* commands의 길이는 1 이상 50 이하입니다.
* commands의 각 원소는 길이가 3입니다.

입출력 예

| **array** | **commands** | **return** |
| --- | --- | --- |
| [1, 5, 2, 6, 3, 7, 4] | [[2, 5, 3], [4, 4, 1], [1, 7, 3]] | [5, 6, 3] |

입출력 예 설명

[1, 5, 2, 6, 3, 7, 4]를 2번째부터 5번째까지 자른 후 정렬합니다. [2, 3, 5, 6]의 세 번째 숫자는 5입니다.  
[1, 5, 2, 6, 3, 7, 4]를 4번째부터 4번째까지 자른 후 정렬합니다. [6]의 첫 번째 숫자는 6입니다.  
[1, 5, 2, 6, 3, 7, 4]를 1번째부터 7번째까지 자릅니다. [1, 2, 3, 4, 5, 6, 7]의 세 번째 숫자는 3입니다.

* 내 풀이

#include <string>

#include <vector>

#include <algorithm>

using namespace std;

vector<int> solution(vector<int> array, vector<vector<int>> commands) {

vector<int> answer;

for(int i=0;i<commands.size();i++) {

vector<int> part;

part.assign(array.begin()+(commands[i][0]-1),array.begin()+(commands[i][1]));

sort(part.begin(),part.end());

answer.push\_back(part[commands[i][2]-1]);

}

return answer;

}

assign 할때, 참조자로 인자가 주어지는데, array+i 이렇게하면안되고, array.begin()+i 이렇게 해야 제대로 입력이 된다. 배열 포인터와는 다르게 생각해야하는 것 같다. array = array.begin()이 아니다.

3.2016년

###### 문제 설명

2016년 1월 1일은 금요일입니다. 2016년 a월 b일은 무슨 요일일까요? 두 수 a ,b를 입력받아 2016년 a월 b일이 무슨 요일인지 리턴하는 함수, solution을 완성하세요. 요일의 이름은 일요일부터 토요일까지 각각 SUN,MON,TUE,WED,THU,FRI,SAT

입니다. 예를 들어 a=5, b=24라면 5월 24일은 화요일이므로 문자열 TUE를 반환하세요.

##### 제한 조건

* 2016년은 윤년입니다.
* 2016년 a월 b일은 실제로 있는 날입니다. (13월 26일이나 2월 45일같은 날짜는 주어지지 않습니다)

#### 입출력 예

| **a** | **b** | **result** |
| --- | --- | --- |
| 5 | 24 | TUE |

* 내 답

#include <string>

#include <vector>

using namespace std;

string solution(int a, int b) {

string answer = "";

int day=b;

vector<string> month{"FRI","SAT","SUN","MON","TUE","WED","THU"};

vector<int> days = {31,29,31,30,31,30,31,31,30,31,30,31};

for(int i=0;i<a-1;i++)

day+=days[i];

answer =month[(day-1)%7];

return answer;

}

1월은 31일부터있고, 7,8월 둘다 31일 나머지는 30일과 번갈아 있다.