

9월 2일 스터디 ~

코드포스 #817 후기 (및 만담)

공식 풀이 (에디토리얼?) ⇒ <https://codeforces.com/blog/entry/106478>

A. Spell Check

- 적용가능 알고리즘: 정렬

B. Colourblindness

- 적용 가능 알고리즘: ?
- 시간복잡도: $O(n)$ per testcase. (라고함)
- 내 풀이: 그냥 막 구현

```
cin >> firstR >> secR;
for (int i = 0; i < n; i++) {
    if (firstR[i] == 'R' || secR[i] == 'R') {
        if (firstR[i] != secR[i]) {
            cout << "NO" << endl;
            return;
        }
    }
}
cout << "YES" << endl;
```

C. Word Game

- 적용 가능 알고리즘: _
- 시간복잡도: $O(n \log n)$ per testcase.

내 풀이

```
manToWords: i 번째 사람이 말한 단어들
wordToScore: 단어의 점수

// 1. manToWords에 3명이 말한 단어를 저장함.
manToWords[i] = {};
```

```

for (int j = 0; j < n; j++) { // n개의 단어
    string word;
    cin >> word;
    manToWords.find(i)->second.push_back(word);

    // 2. 3명이 말한 단어를 저장할 때 입력으로 들어오는 단어가
    // wordToScore에 이미 있다면 점수 갱신, 없으면 3점으로 저장
    map<string, int>::iterator it = wordToScore.find(word);
    if (it != wordToScore.end()) {
        if (it->second == 3) {
            it->second = 1;
        } else {
            it->second = 0;
        }
    } else {
        wordToScore[word] = 3;
    }
}

```

D. Line

내 풀이

우선순위 큐로 켈 많이 차이나는 애들부터 원래 스코어에 더해주었다

▼ bestRes 배열

```

// 각 위치에서 보이는 최대 사람수 (그래봤자 둘중에 하나. 왼쪽 또는 오른쪽)

int bestRes[2 * 100000] = {};
for (int i = 0; i < n / 2 + 1; i++) {
    if (bestRes[i] != 0) break;
    bestRes[i] = bestRes[n - 1 - i] = n - 1 - i;
}

```

```

int currRes[2 * 100000] = {};
long long total = 0;

priority_queue<long long> pQueue = {};

for (int i = 0; i < n; i++) {
    if (pos[i] == 'L') {
        currRes[i] = i;
    } else {
        currRes[i] = n - 1 - i;
    }

    if (bestRes[i] != currRes[i]) {
        pQueue.push(bestRes[i] - currRes[i]);
    } // 굳이 bestRes를 배열에 저장하지는 않아도 되었을 것같네요
}

```

```

// int bestRes = max(i, n-1-i));
// if (currRes[i] != bestRes) {
//     pQueue.push(bestRes-currRes[i]);
// }

total += currRes[i];
}

```

E. Counting Rectangles

내 풀이

```

// 높이가 i, 너비가 j인 영역에 들어가는 직사각형 넓이의 합 계산
for (int i = 2; i <= 1000; i++) {
    for (int j = 2; j <= 1000; j++) {
        // 밑에서부터 순서대로 계산하므로 이전에 계산됐는지 체크하지 않고 바로 대입할 수 있음
        dp[i][j] = dp[i - 1][j] + dp[i][j - 1] - dp[i - 1][j - 1] + area[i - 1][j - 1];
    }
}

```

- 입력의 크기를 확인해야한다. (넓이 값을 저장하는 배열을 int로 선언하니 범위를 넘는 수가 들어갔을 때 음수가 되어 wrong answer가 되었다.)
- 그리고 배열을 함수 내에서 선언하니까 segmentation fault 가 떠서, 전역으로 배열을 선언하고 각 테스트 케이스마다 배열을 초기화하여 사용했다.

F. L-Shapes

문제만 열심히 읽었음

G. Even-Odd XOR

문제를 열심히 읽었음