

**문제 링크주소 :** <https://programmers.co.kr/learn/courses/30/lessons/42862>

**풀이**

function solution(*n*, *lost*, *reserve*) {

    var answer = 0;

    function same(){

        for(var i=0;i<*lost*.length; i++){

            for(var j=0;j<*reserve*.length; j++){

                if(*lost*[i]==*reserve*[j]){

*lost*.splice(i,1);

*reserve*.splice(j,1);

                same();

                }

            }

        }

    }

    function low(){

        for(var i=0;i<*lost*.length; i++){

            for(var j=0;j<*reserve*.length; j++){

                if(*lost*[i]-1 == *reserve*[j]){

*lost*.splice(i,1);

*reserve*.splice(j,1);

                low();

                }

            }

        }

    }

    function high(){

        for(var i=0;i<*lost*.length; i++){

            for(var j=0;j<*reserve*.length; j++){

                if(*lost*[i]+1 == *reserve*[j]){

*lost*.splice(i,1);

*reserve*.splice(j,1);

                high();

                }

            }

        }

    }

    same();

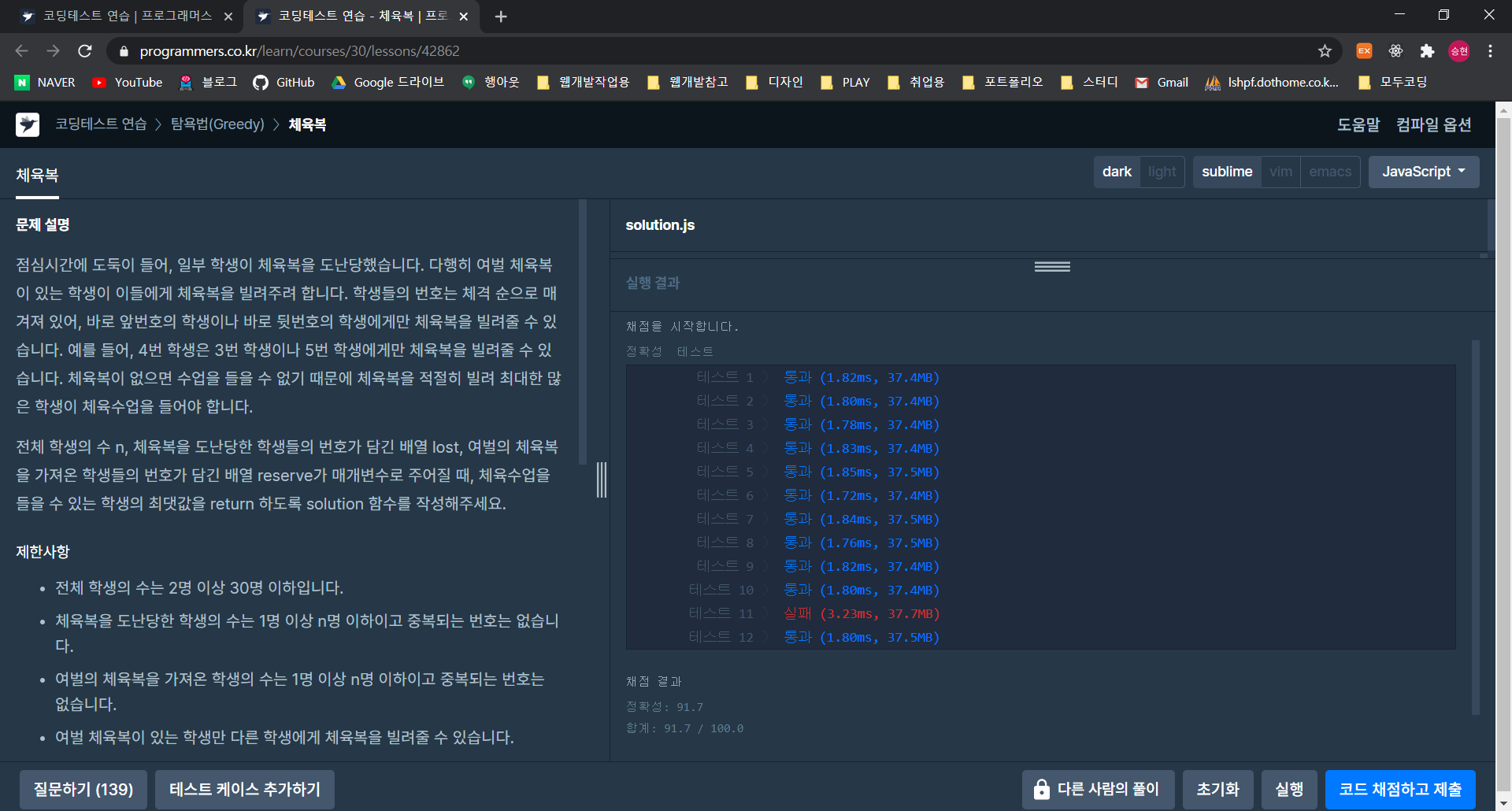
    low();

    high();

    answer = *n*-*lost*.length;

    return answer;

}



다른사람의 풀이

function solution(n, lost, reserve) {

    let answer = n;

*// 두 배열에 중복으로 존재하는 학생들은 양쪽 목록에서 제거*

    lost.map((stdNum) => {      *//lost배열에서 map으로 새로 배열 생성*

        if(reserve.includes(stdNum)){ *// reserve배열에 lost배열의 stdNum을 포함하는지 확인*

*//lost중에 중복되는 학생들을 제외해서 새로배열 작성*

            lost = lost.filter((lostStdNum) => stdNum !== lostStdNum);

*//reserve중에 중복되는 학생들을 제외해서 새로배열 작성*

            reserve = reserve.filter((reserveStdNum) => stdNum !== reserveStdNum);

        }

    });

*// 앞부터 확인해서 여분 가져온 학생 있으면 빌리기*

    lost.map((stdNum) => {

        let beforeStdNum = stdNum - 1;

        let afterStdNum = stdNum + 1;

        if(reserve.includes(beforeStdNum)){ *//lost배열에있는값보다 1작은 학생이 있으면*

*//lost중에 해당 학생들을 제외해서 새로배열 작성*

            lost = lost.filter((lostStdNum) => stdNum !== lostStdNum);

*//reserve중에 해당 학생들을 제외해서 새로배열 작성*

            reserve = reserve.filter((reserveStdNum) => beforeStdNum !== reserveStdNum);

        }else if(reserve.includes(afterStdNum)){

            lost = lost.filter((lostStdNum) => stdNum !== lostStdNum);

            reserve = reserve.filter((reserveStdNum) => afterStdNum !== reserveStdNum);

        }

    });

    return answer - lost.length;

}

function solution(*n*, *lost*, *reserve*) {

    return *n* - *lost*.filter(*a* => {

        const b = *reserve*.find(*r* => Math.abs(*r*-*a*) <= 1)

*// reserve배열 중에서  reserve-lost의 절대값이 1보다 작거나 같은*

*// 즉, reserve와 lost의 차이가 1 이내인 값들만 추려서 b라는 변수에 배열형태로 만들어 담는다.*

        if(!b) return true

*//lost중에서 이거에 해당하지 않는 애들만 추려낸다*

*reserve* = *reserve*.filter(*r* => *r* !== b)

*//reser도 해당하지 않는 애들만 추려낸다.*

    }).length

}