ĐẠI HỌC BÁCH KHOA HÀ NỘI

**TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**

----- 🙡 🕮 🙣 -----



**BÁO CÁO PROJECT I**

***Báo cáo tuần 9***

Giảng viên: **Ngô Lam Trung**

Sinh viên thực hiện: Nguyễn Quang Long

Mã số sinh viên: 20194320

**Hà Nội, năm 2021**

Nội dung đã hoàn thành:

Từ câu 150 đến câu 154

Phần Intro: từ câu 11 đến câu 25

**11. isLucky**

a. Đề bài

1 số được gọi là lucky nếu tổng các chữ số nửa đầu bằng tổng các chữ số nửa sau. Cho 1 số, kiểm tra xem số đó có lucky không

b. Giải pháp

Tính tổng các chữ số nửa đầu và nửa sau rồi so sánh

c. Code

bool solution(int n) {

    string n\_string = to\_string(n);

    int sum1 =0, sum2=0;

    int len = n\_string.length();

    for(int i =0; i<len/2;i++){

        sum1+= (int) n\_string[i]-'0';

        sum2+= (int) n\_string[len-1-i]-'0';

    }

    if(sum1==sum2) return true;

    else return false;

}

**12. Sort by Height**

a. Đề bài

Cho 1 mảng gồm các số nguyên dương và các số -1, sắp xếp mảng theo thứ tự tăng dần nhưng vẫn giữ nguyên vị trí các số -1

b. Giải pháp

Sử dụng sắp xếp chọn kết hợp thêm điều kiện số nhỏ hơn khác -1

c. Code

vector<int> solution(vector<int> a) {

    for(int i =0 ;i<a.size();i++){

        for(int j=i+1;j<a.size();j++){

            if(a[i]>a[j] && a[j]!=-1) swap(a[i],a[j]);

        }

    }

    return a;

}

**13. Reverse in parentheses**

a. Đề bài

Viết hàm đảo ngược các xâu con nằm trong ( ) của một xâu đã cho (có thể có nhiều ()).

Ví dụ For inputString = "foo(bar(baz))blim", the output should be  
solution(inputString) = "foobazrabblim".

b. Giải pháp

Duyệt lần lượt từ đầu xâu, nếu gặp ký tự ‘(‘ thì push vị trí của ký tự đó vào stack, nếu gặp ‘)’ thì đảo ngược xâu con từ vị i nằm ở đỉnh stack đến vị trí của ký tự ‘)’.

Sau đó loại bỏ các ký tự ‘(‘ và ‘)’ của xâu đã cho.

c. Code

string solution(string str) {

     stack<int> st;

     int len = str.length();

    for (int i = 0; i < len; i++) {

        if (str[i] == '(') {

            st.push(i);

        }

        else if (str[i] == ')') {

            reverse(str.begin() + st.top() + 1,

                    str.begin() + i);

            st.pop();

        }

    }

    string res = "";

    for (int i = 0; i < len; i++) {

        if (str[i] != ')' && str[i] != '(')

            res += (str[i]);

    }

    return res;

}

**14. Alternating Sums**

a. Đề bài

Cho 1 mảng các số nguyên dương, tính tổng các phần tử ở vị trí chẵn và các phần tử ở vị trí lẻ

b. Giải pháp

Duyệt 2 vòng lặp với khoảng cách là 2

c. Code

vector<int> solution(vector<int> a) {

    int sum1=0, sum2=0;

    for(int i=0;i<a.size();i+=2){

        sum1+=a[i];

    }

    for(int i=1;i<a.size();i+=2){

        sum2+=a[i];

    }

    return {sum1,sum2};

}

**15. Add Border**

a. Đề bài

Cho 1 ma trận hình chữ nhật gồm các ký tự, hãy thêm đường bao vào ma trận đó bằng các ký tự \*

b. Giải pháp

Tạo 1 mảng mới rồi thêm lần lượt đường bao trên, dưới và trái , phải vào mảng đã cho

c. Code

vector<string> solution(vector<string> picture) {

    int row = picture.size();

    int col = picture.at(0).size();

    vector<string> ans;

    string add ="\*";

    for(int i=1;i<col+2;i++){

        add.append("\*");

    }

    ans.push\_back(add);

    for(int i=0;i<row;i++){

        ans.push\_back("\*"+picture.at(i)+"\*");

    }

    ans.push\_back(add);

    return ans;

}

**16. Are Similar?**

a. Đề bài

2 mảng được gọi là similar nếu 1 mảng có thể chứa 1 mảng khác nếu đổi chỗ ít nhất 1 cặp phần tử trong 1 trong 2 mảng. Cho 2 mảng, kiểm tra chúng có similar không

b. Giải pháp

Kiểm tra xem có bao nhiêu vị trí tương ứng khác nhau trong 2 mảng, nếu =0 thì trả về true, nếu =2 thì sắp xếp lại 2 mảng, nếu 2 mảng bằng nhau thì trả về true.

Ngược lại trả về false

c. Code

bool solution(vector<int> a, vector<int> b) {

    int count=0;

    for(int i=0;i<a.size();i++){

        if(a[i]!=b[i]) count++;

    }

    if(count==0) return true;

    if(count==2) {

    sort(a.begin(),a.end());

    sort(b.begin(),b.end());

    if(a==b) return true;

}

return false;

}

**17. Array Change**

a. Đề bài

Cho 1 mảng, tính xem cần cộng thêm vào các phần tử của mảng tất cả ít nhất bao nhiêu để được 1 dãy tăng dần?

b. Giải pháp

Duyệt lần lượt các phần tử, nếu phần tử đứng ngay sau nhỏ hơn hoặc bằng phần tử đứng trước thì tính toán lượng cần cộng thêm tối thiểu rồi cộng vào phần tử đứng sau và cộng thêm vào tổng tất cả.

c. Code

int solution(vector<int> inputArray) {

    int moves = 0;

    for(int i =1 ;i<inputArray.size();i++){

        if(inputArray[i]<=inputArray[i-1]){

           int move = inputArray[i-1]+1-inputArray[i];

            inputArray[i] += move;

            moves += move;

        }

    }

    return moves;

}

**18. Palindrome Rearranging**

a. Đề bài

Cho 1 xâu, kiểm tra xem có thể sắp xếp lại các ký tự để trở thành xâu palindrome không

b. Giải pháp

Tính xem mỗi ký tự trong xâu xuất hiện bao nhiêu lần, nếu có > 1 ký tự xuất hiện lẻ lần thì trả về false, ngược lại true

c. Code

bool solution(string inputString) {

    int arr[26]= {0};

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

        arr[inputString[i]-'a']++;

    }

    int count = 0;

    for(int i=0;i<26;i++){

        if(arr[i]&1==1) count ++;

        if(count>1) return false;

    }

    return true;

}

**19. Are equally Strong**

a. Đề bài

2 người được gọi là khỏe như nhau nếu tay yếu nhất và tay mạnh nhất đều khỏe như nhau. Kiểm tra xem 2 người có khỏe như nhau không

b. Giải pháp

Kiểm tra từng cặp tay yếu và tay khỏe của 2 người

c. Code

bool solution(int yourLeft, int yourRight, int friendsLeft, int friendsRight) {

    if(max(yourLeft,yourRight)==max(friendsLeft,friendsRight) && min(yourLeft,yourRight)==min(friendsLeft,friendsRight)) return true;

    return false;

}

**20. Array maximal adjacent difference**

a. Đề bài

Tìm khoảng cách lớn nhất giữa 2 cặp phần tử liên tiếp trong mảng

b. Giải pháp

Duyệt lần lượt các phần tử và tính khoảng cách rồi cập nhật biến max

c. Code

int solution(vector<int> inputArray) {

    int max = -1;

    for(int i=0;i<inputArray.size()-1;i++){

        if(max < abs(inputArray[i]-inputArray[i+1]))

        max = abs(inputArray[i]-inputArray[i+1]);

    }

    return max;

}

**21. Is Ipv4 address**

a. Đề bài

Kiểm tra xem xâu đã cho có phải địa chỉ Ipv4 không

b. Giải pháp

Tách xâu đã cho bởi các ký tự ‘.’, nếu xâu có khác 4 phần tử thì trả về false. Ngược lại duyệt lần lượt các phần tử và kiểm tra theo điều kiện của ipv4

c. Code

function solution(inputString) {

    let numbers = inputString.split(".");

    if (numbers.length !== 4) return false;

    for (i = 0; i < 4; i++) {

        if (numbers[i] !== "0") {

            if( numbers[i][0] == 0) return false;

            if (numbers[i] === "") return false;

            let number = Number(numbers[i]) || -1 ;

            if (number < 0 || number > 255) {

             return false;

            }

        }

    }

    return true;

}

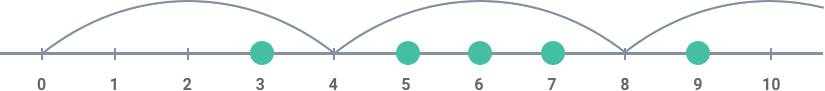
**22. Avoid Obstacles**

a. Đề bài

Cho 1 mảng các số nguyên được coi là vị trí các chướng ngại vật, tính xem cần sử dụng các bước nhảy ít nhất bao nhiêu để không gặp chướng ngại vật, giả sử bắt đầu từ 0

For inputArray = [5, 3, 6, 7, 9], the output should be  
solution(inputArray) = 4.

Check out the image below for better understanding:



b. Giải pháp

Thử các bước nhảy từ 2 đến phần tử lớn nhất trong mảng, nếu như không gặp CNV thì kết thúc ngay vòng lặp, ngược lại nếu không có giá trị thỏa mãn thì trả về bước nhảy min = max +1

c. Code

int solution(vector<int> inputArray) {

    int max = inputArray[0];

    for(int i = 1 ; i<inputArray.size();i++){

        if(max < inputArray[i]) max = inputArray[i];

    }

    for(int i  = 2 ; i<max; i++){

        bool meet = false;

        for(int j =0;j<inputArray.size();j++){

            if(inputArray[j]%i==0) {

                meet = true;

                break;

            }

        }

        if(!meet) return i;

    }

    return max+1;

}

**23. Box Blur**

a. Đề bài

Cho 1 ma trận hình vuông nxn , hãy thay thế các ma trận con 3x3 bằng giá trị trung bình các phần tử của chúng

b. Giải pháp

Duyệt lần lượt các ma trận con 3x3 rồi tính giá trị trung bình và lưu vào mảng kết quả. Kích thước của mảng kết quả là (n-2)x(n-2)

c. Code

vector<vector<int>> solution(vector<vector<int>> image) {

    vector<vector<int>> ans;

    for(int i=1;i<image.size()-1;i++){

        vector<int> sub;

        for(int j=1;j<image[0].size()-1;j++){

            int element = (image[i-1][j-1]+image[i-1][j]+image[i-1][j+1]+image[i][j-1]+image[i][j]+image[i][j+1]+image[i+1][j-1]+image[i+1][j]+image[i+1][j+1])/9;

            sub.push\_back(element);

        }

        ans.push\_back(sub);

    }

    return ans;

}

**24. Minesweeper**

a. Đề bài

Cho 1 ma trận bool là vị trí bom của game minesweeper, hãy đưa ra ma trận với mỗi phần tử là số mìn xung quanh của từng vị trí

b. Giải pháp

Dùng 1 ma trận để lưu kết quả với giá trị khởi tạo bằng 0, duyệt ma trận đã cho, với mỗi vị trí xuất hiện bom thì tăng các phần tử xung quanh của ma trận kq lên 1

c. Code

void check(vector<vector<int>> &res ,int i, int j, int row, int col){

    if(i>=0&&i<=(row-1)&&j>=0&&j<=(col-1)) res[i][j]++;

}

vector<vector<int>> solution(vector<vector<bool>> matrix) {

    vector<vector<int>> res;

    int row = matrix.size();

    int col = matrix[0].size();

    vector<int> sub;

    for(int i=0;i<col;i++){

        sub.push\_back(0);

    }

    for(int i=0;i<row;i++){

        res.push\_back(sub);

    }

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

        for(int j=0;j<matrix[0].size();j++){

            if(matrix[i][j]) {

                check(res,i-1,j-1,row,col);

                check(res,i-1,j,row,col);

                check(res,i-1,j+1,row,col);

                check(res,i,j-1,row,col);

                check(res,i,j+1,row,col);

                check(res,i+1,j-1,row,col);

                check(res,i+1,j+1,row,col);

                check(res,i+1,j,row,col);

            }

        }

    }

    return res;

}

**25. Array Replace**

a. Đề bài

Cho 1 mảng, hãy đổi giá trị các phần tử = elemtoReplace với substitutionElem

b. Giải pháp

Duyệt mảng, nếu có phần tử bằng elemtoReplace thì thay thế bằng substitutionElem

c. Code

vector<int> solution(vector<int> inputArray, int elemToReplace, int substitutionElem) {

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

        if(inputArray[i]==elemToReplace) inputArray[i] = substitutionElem;

    }

    return inputArray;

}