Министерство цифрового развития, связи и массовых коммуникаций

Ордена Трудового Красного Знамени федеральное государственное бюджетное

образовательное учреждение высшего образования

«Московский технический университет связи и информатики»

Кафедра Математической кибернетики и информационных технологий

Отчёт по практическим заданиям

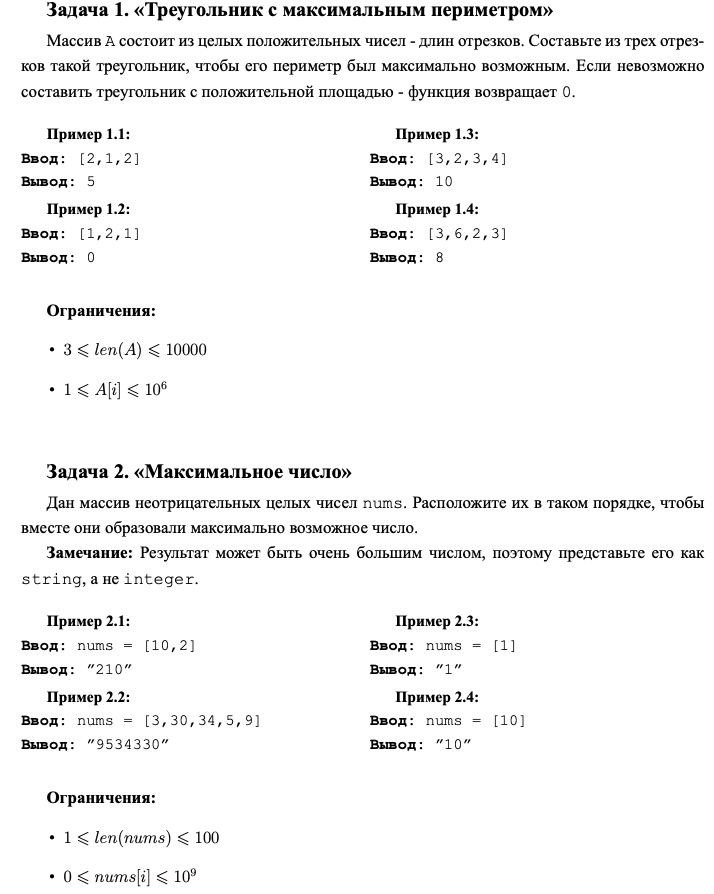
по дисциплине «Структуры и алгоритмы обработки данных»

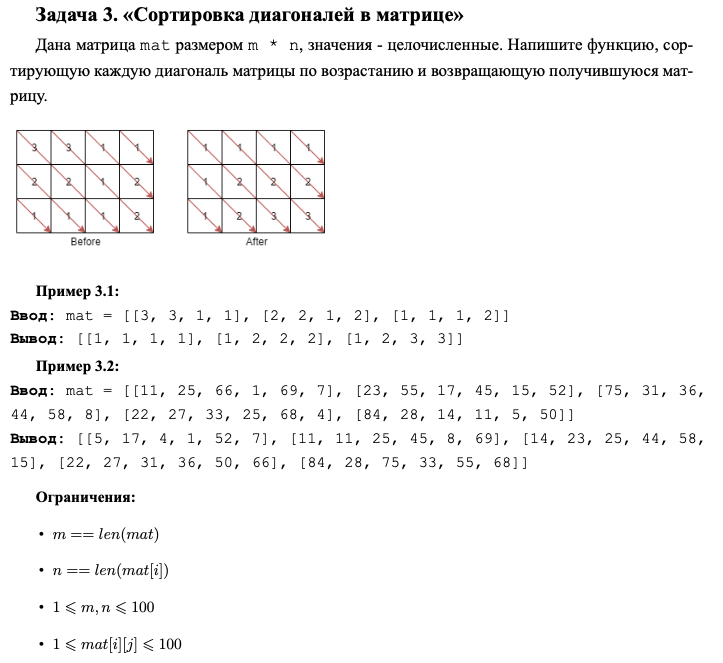
Выполнил: студент группы БВТ1902

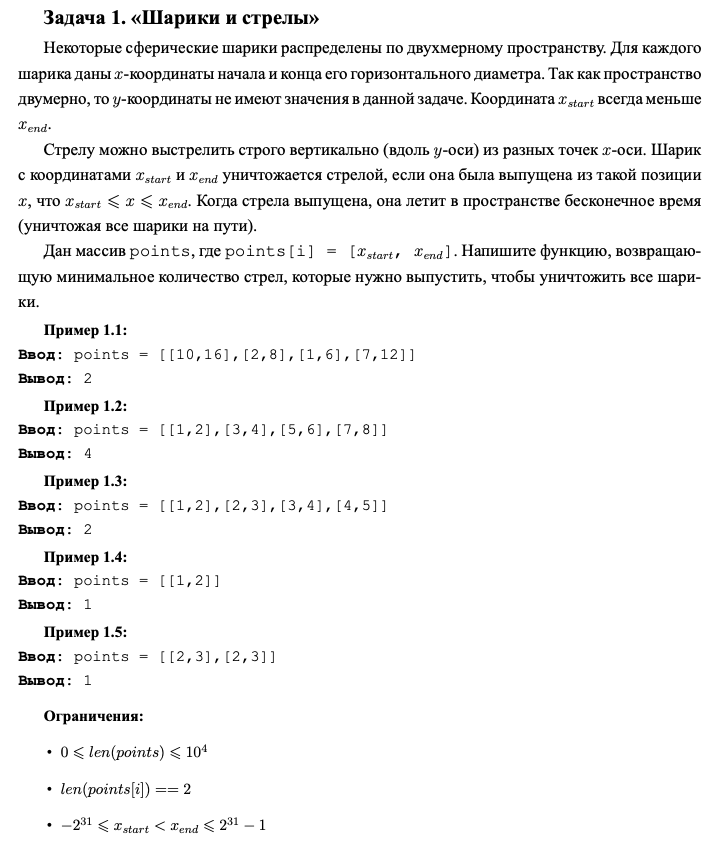
Холмовский Олег Васильевич

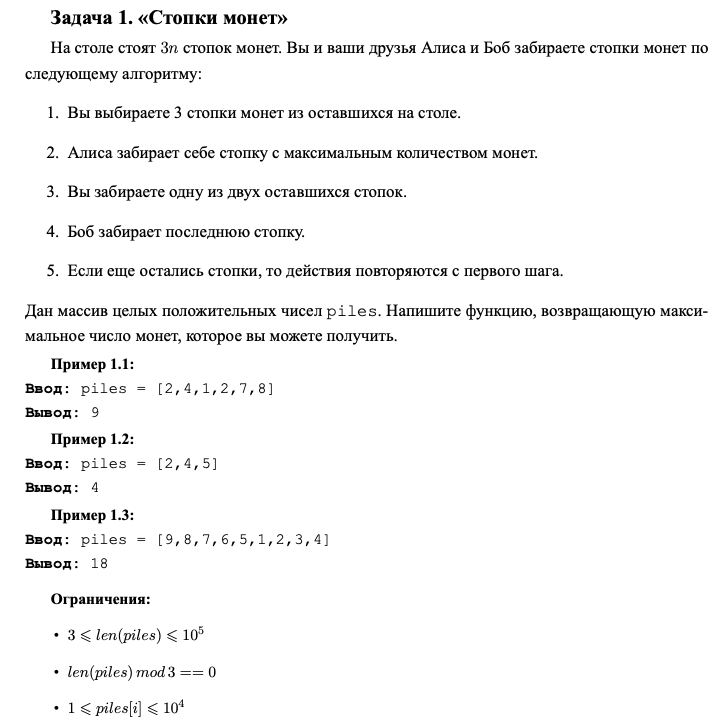
Москва

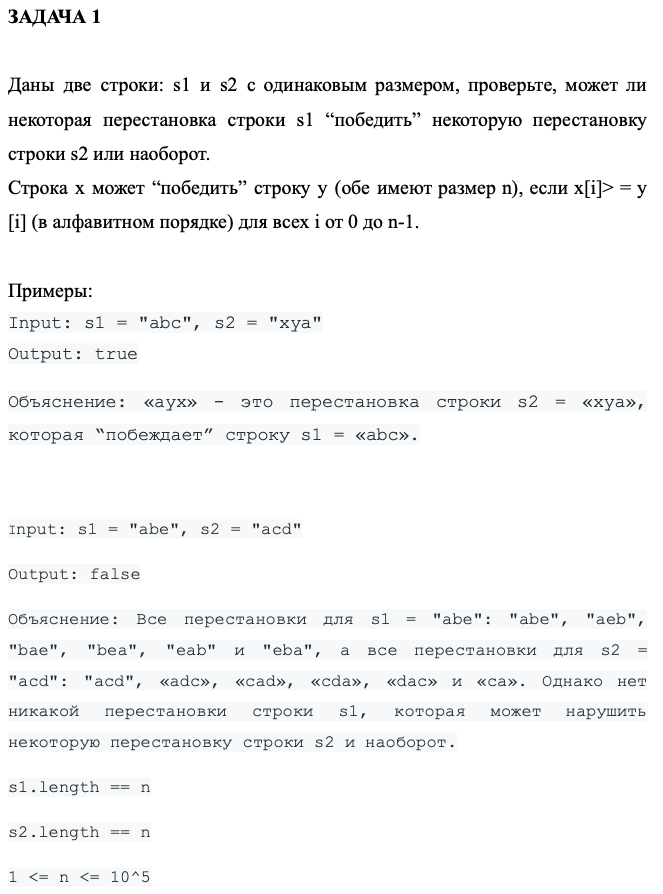
2021

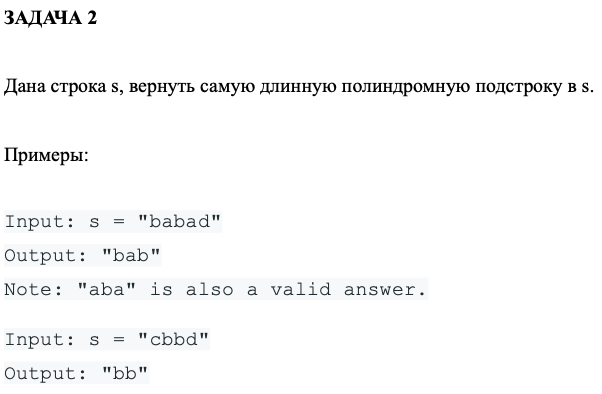
Задание

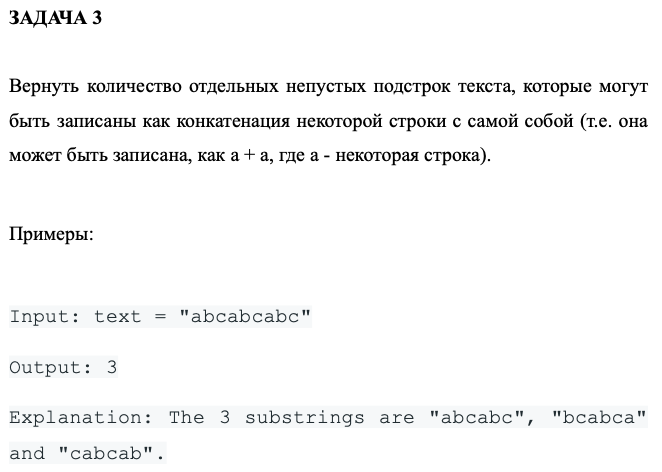












Код программы

Задачи треугольник с максимальным периметром, максимальное число, сортировка диагоналей в матрице:

const getMaxP = (arr) => {

sortedArr = arr.sort((a, b) => b - a);

let maxP = 0;

*for* (let i = 0; i < sortedArr.length - 2; i++) {

*if* (sortedArr[i] < sortedArr[i + 1] + sortedArr[i + 2]) {

maxP = Math.max(maxP, sortedArr[i] + sortedArr[i + 1] + sortedArr[i + 2]);

}

}

*if* (maxP) console.log(maxP);

*else* console.log('0');

};

console.log('getMaxP:');

console.log('[21, -11, 10, -5, 20]');

getMaxP([21, -11, 10, -5, 20]);

const getLargestNumber = (arr) => {

arr.sort((first, second) => {

let firstsecond = '' + first + second;

let secondfirst = '' + second + first;

*return* firstsecond > secondfirst ? -1 : 1;

});

let largestNumber = arr.join('');

console.log(largestNumber);

};

console.log('getLargestNumber:');

console.log('[3, 30, 34, 5, 9, 1, 123, 123, 4]');

getLargestNumber([3, 30, 34, 5, 9, 1, 123, 123, 4]);

const diagSort = (mat) => {

let y = mat.length,

x = mat[0].length - 1,

diag = *new* Array(y);

let k;

*for* (let i = 2 - y; i < x; i++) {

k = 0;

*for* (let j = 0; j < y; j++)

*if* (i + j >= 0 && i + j <= x) diag[k++] = mat[j][i + j];

diag.sort((a, b) => a - b);

k = 0;

*for* (let j = 0; j < y; j++)

*if* (i + j >= 0 && i + j <= x) mat[j][i + j] = diag[k++];

}

*for* (let i = 0; i < mat.length; i++) {

console.log(mat[i]);

}

};

console.log('diagSort:');

console.log(`[[3, 3, 1, 1],

[2, 2, 1, 2],

[1, 1, 1, 2]]

`);

diagSort([

[3, 3, 1, 1],

[2, 2, 1, 2],

[1, 1, 1, 2],

]);

Задача шарики и стрелы:

const problemBalloons = (points) => {

*if* (points.length == 0) *return* 0;

points.sort((a, b) => a[1] - b[1]);

let arrowPos = points[0][1];

let count = 1;

*for* (let i = 1; i < points.length; i++) {

*if* (arrowPos >= points[i][0]) {

*continue*;

}

count++;

arrowPos = points[i][1];

}

*return* count;

};

console.log('problemBalloons:');

console.log(`[[10, 16],

[2, 8],

[1, 6],

[7, 12]]`);

console.log(

problemBalloons([

[10, 16],

[2, 8],

[1, 6],

[7, 12],

])

);

Задача стопки монет:

const problemCoins = (arr) => {

let result = 0;

arr.sort((a, b) => a - b);

*for* (let i = arr.length / 3; i < arr.length; i = i + 2) {

result += arr[i];

}

*return* result;

};

console.log('problemCoins:');

console.log(`[2, 4, 1, 2, 7, 8]`);

console.log(problemCoins([2, 4, 1, 2, 7, 8]));

Задачи на строки-2:

const winString = (str1, str2) => {

*if* (str1.length === 0 || str2.length === 0) {

*return* `empty string`;

} *else* *if* (str1.length !== str2.length) {

*return* `different length`;

}

let arrChar1 = str1.split('').sort();

let arrChar2 = str2.split('').sort();

*if* (

arrChar1.every((el, idx) => el <= arrChar2[idx]) ||

arrChar2.every((el, idx) => el <= arrChar1[idx])

) {

*return* true;

}

*return* false;

};

console.log('winString:');

console.log(`'abe', 'xya'`);

console.log(winString('abe', 'xya'));

const isPolindrome = (string) => {

*return* string === string.split('').reverse().join('');

};

const longestPolindrome = (string) => {

*if* (string.length === 0) {

*return* `it is empty`;

}

let longest = '';

*for* (let i = 0; i < string.length; i++) {

*for* (let j = 0; j < string.length; j++) {

let substr = string.slice(i, j);

*if* (isPolindrome(substr)) {

*if* (longest.length < substr.length) {

longest = substr;

}

}

}

}

*if* (longest.length === 0) {

*return* `no polindrome`;

}

*return* longest;

};

console.log('longestPolindrome:');

console.log('babad');

console.log(longestPolindrome('babad'));

const subCount = (string) => {

let result = [];

let length = string.length;

*for* (let i = 0; i < length; i++) {

*for* (let j = i + 1; j < length + 1; j++) {

let sub = string.slice(i, j);

*for* (let i = 0; i < sub.length; i++) {

*if* (sub.slice(0, i) === sub.slice(i, sub.length)) {

result.push(sub);

*continue*;

}

}

}

}

*return* *new* Set(result).size;

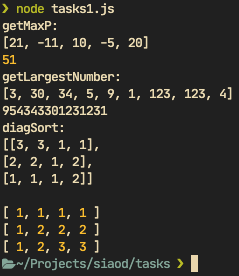
};

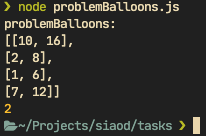
console.log('subCount:');

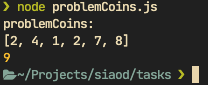
console.log('abcabcabc');

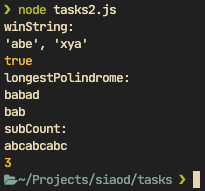
console.log(subCount('abcabcabc'));

Cнимки экрана работы программы









Вывод

В ходе выполнения практических задач, мной были применены навыки, полученные в течения курса, что позволило мне реализовать оптимальное решение.