## МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЕ РФ

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

## “ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ”

Кафедра «Программное обеспечение вычислительной техники и автоматизированных систем»

Лабораторная работа №3

По дисциплине: «Эвристические методы и алгоритмы» На тему «Теория расписаний»

Выполнил :

Студент группы ВМО31

Оганесьянц К.П.

Проверил:

Кобак В.Г.

Ростов-на-Дону 2022 г.

# Введение

Задачи проектирования и управления в системах, для которых необходимо распределение работы между параллельно работающими разнородными вычислительными устройствами занимают значимое место в теории построения расписаний. Практическая актуальность таких задач определяется существенными возможностями экономии машинного времени и вытекающими функциональными и эксплуатационными преимуществами.

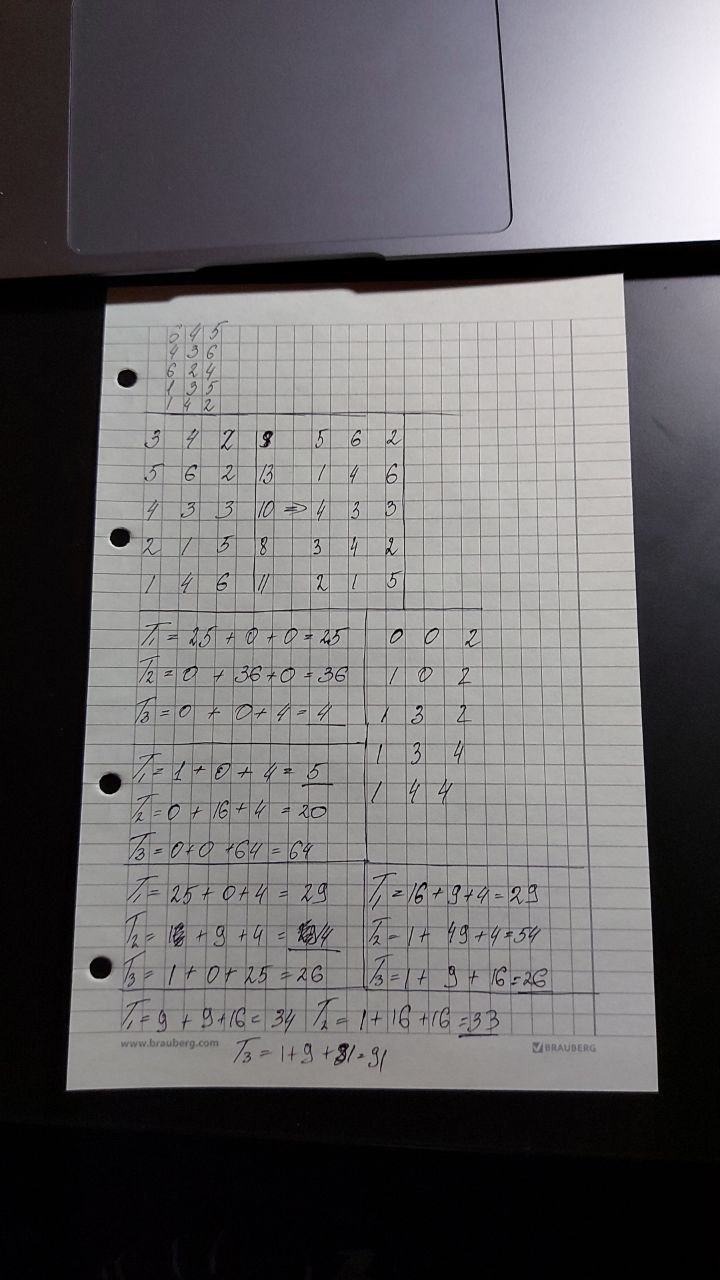
Теоретическая сложность нахождения наилучшего распределения связана с необходимостью решения экстремальных задач комбинаторного типа, требующих больших вычислительных ресурсов, так что эффект от нахождения близкого к оптимальному, с точки зрения времени выполнения, распределения может быть сведен на нет затратами на его получение.

В настоящем руководстве приводятся методы получения расписаний, приводящие к небольшим затратам на вычисление за счет отказа от получения оптимального решения, но в тоже время позволяющие найти приемлемое решение, близкое к оптимальному.

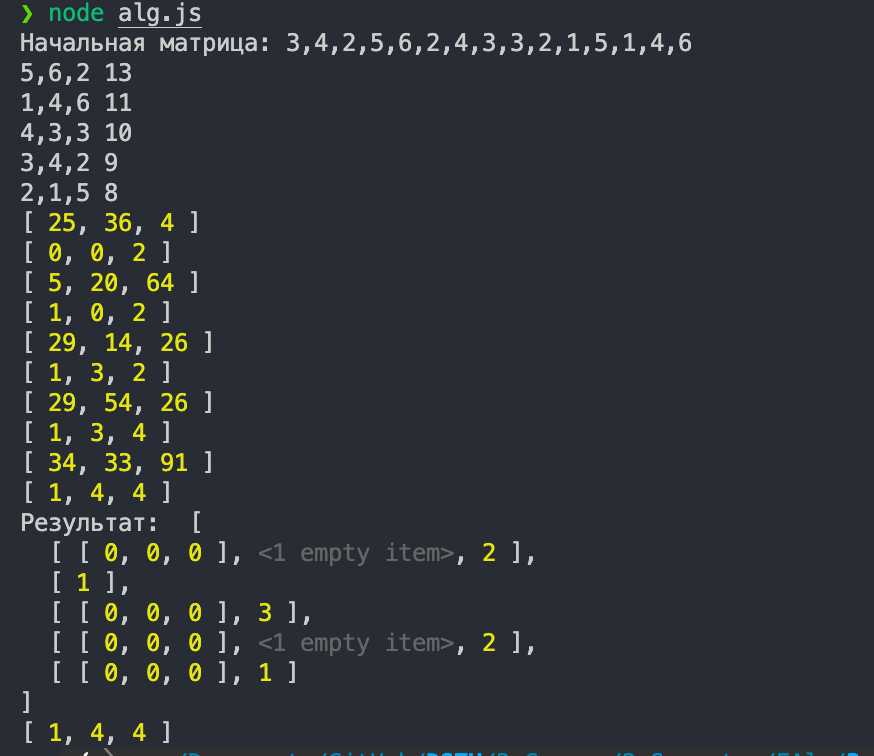
# Постановка задачи

Имеется  независимых работ , которые необходимо распределить на  параллельно работающих разнородных устройств  по критерию .Получение оптимального распределения в такой постановке приводит к громоздким вычислениям, требующим значительного времени машинного счета, поэтому цель – продемонстрировать алгоритмы, с помощью которого можно находить с малыми затратами достаточно приемлемое решение.

1. **Результаты тестирования программы**



Для тестирования программы были заданы параметры.



# Код программы

function randomIntFromInterval(min, max) { *// min and max included*

return Math.floor(Math.random() \* (max - min + 1) + min)

}

const calculate = (n, m, t1, t2) => {

const array = [];

for (let i = 0; i < m; i++) {

let array2 = []

for (let j=0; j < n; j++) {

array2.push(randomIntFromInterval(t1, t2))

}

array.push(array2)

}

console.log(`Начальная матрица: ${array}`);

const sorted = array.sort(function(a, b){return b.reduce(function(a, b){

return a + b;

}, 0) - a.reduce(function(a, b){

return a + b;

}, 0)});

array.map(el => {

console.log(`${el.toString()} ${el.reduce((el,acc) => acc + Number(el),0)}`)

})

let processes = Array(n).fill(0);

let result = [];

let sumArray = [];

let sum = 0;

let minIndex;

for(let x=0; x < m; x++) {

for (let element=0;element<n;element++) {

for (let y=0; y<n; y++) {

if(element !== y) {

sum = sum + (processes[y]\*processes[y])

} else {

sum = sum+((processes[y]+sorted[x][y])\*(processes[y]+sorted[x][y]))

}

}

sumArray.push(sum);

sum = 0;

}

console.log(sumArray)

minIndex = sumArray.indexOf(Math.min(...sumArray))

result.push([Array(processes.length).fill(0)])

result[x][minIndex] = sorted[x][minIndex]

processes[minIndex] += sorted[x][minIndex]

sumArray = []

console.log(processes)

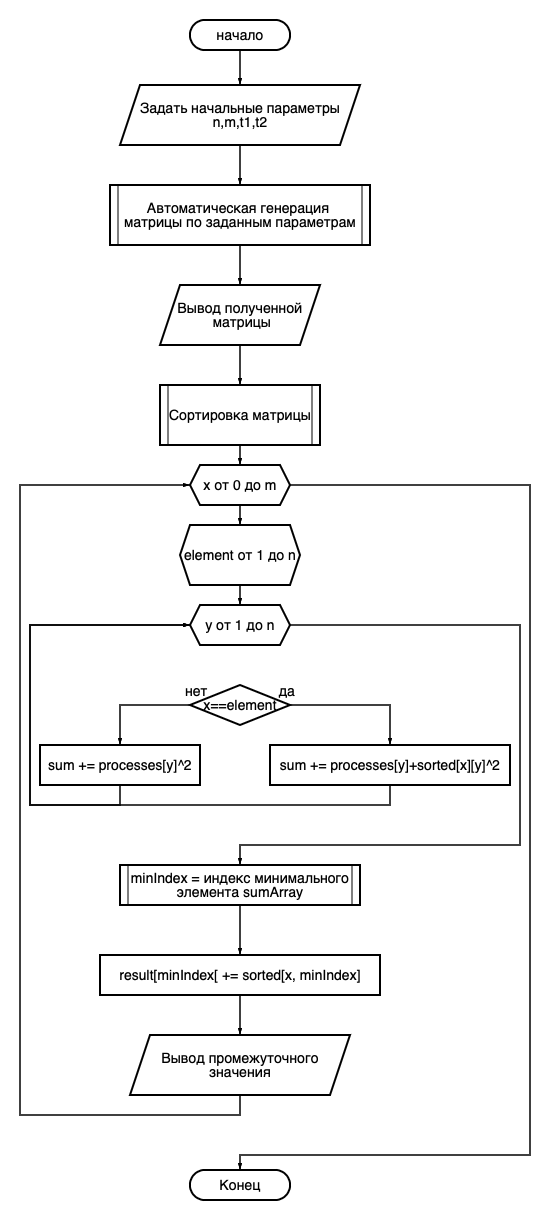
}

console.log('Результат: ', result)

return processes;

}

# Блок схема

****

1. **Вывод**

Алгоритм построения расписания с методом наименьших квадратов достаточно эффективен по скорости поиска приемлемого по точности решения.

# Литература

1. Коффман Э.Г. “Теория расписания и вычислительные машины” – M.: “Наука”, 1987
2. Романовский И.В. “Алгоритмы решения экстремальных задач” – М.: “Наука”, 1977
3. Пашкеев С.Д., Минязов Р.И., Могилевский В.Д. “Машинные методы оптимизации в технике связи” – М.: “Связь”, 1976.