# Лабораторная работа № 1
# «Простейший пример оптимизации»
# (4 часа)

*Цель работы: изучение различных методов решения простейших задач оптимизации.*

## 1 КРАТКИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

### 1.1 Постановка задачи оптимизации

При изготовлении некоторого изделия используются пять рабочих мест в различных местах завода. Последовательность обработки не имеет значения. Время обработки на всех рабочих местах одинаково и равно 1 мин. Время перемещения изделия от одного рабочего места к другому различно и определяется таблицей 1. Стоит задача минимизации времени изготовления изделия. В качестве ограничения стоит требование забирать изделие из того рабочего места, на которое оно поступило первоначально. Таким образом:

$T_{\text{изготовления}} = T_{\text{обработки}} + T_{\text{транспортировки}}$.

Учитывая, что $T_{\text{обработки}} = const$, то задача сводится к минимизации времени транспортировки.

Таблица 1 – Время транспортировки

| | $P_1$ | $P_2$ | $P_3$ | $P_4$ | $P_5$ |
|---|---|---|---|---|---|
| $P_1$ | 0 | И(10) | в(3) | а(1) | н(15) |
| $P_2$ | 10 | 0 | о(16) | в(3) | П(17) |
| $P_3$ | 3 | 16 | 0 | е(6) | т(20) |
| $P_4$ | 1 | 3 | 6 | 0 | р(18) |
| $P_5$ | 15 | 17 | 20 | 18 | 0 |

В общем случае для определения минимального времени транспортировки необходимо выполнить полный перебор всех возможных вариантов. Однако для больших значений числа рабочих мест данный метод неприемлем. Поэтому используют другие подходы, которые имеют меньшую вычислительную сложность. Одним из таких подходов является «жадный» алгоритм. Суть алгоритма заключается в следующем. В первой строке находим минимальный элемент (в примере он равен 1) и переходим с строке, которая имеет номер столбца, в котором расположен минимальный элемент (в примере – 4). Затем в четвертой строке ищем минимальный элемент, при этом игнорируем столбцы с номерами строк, на которых мы уже побывали, то есть

min (3, 6, 18)=3.

Данные действия повторяются до тех пор, пока не будут пройдены все строки. В примере:

$P_1 – (1) – P_4 – (3) – P_2 – (16) – P_3 – (20) – P_5 – (15) – P_1.$

$1 + 3 + 16 + 20 + 15 = 55.$

Таким образом, используя «жадный» алгоритм, получили:

$T_{\text{транспортировки}}$=55 мин.

Для примера рассмотрим еще одну последовательность транспортировки

$P_1 – (10) – P_2 – (16) – P_3 – (6) – P_4 – (18) – P_5 – (15) – P_1.$

$10 + 16 + 6 + 18 + 15 = 65.$

Таким образом, применение «жадного» алгоритма позволяет найти некоторое оптимальное (квазиоптимальное) решение данной задачи. Однако следует помнить, что данный алгоритм не гарантирует нахождение минимума.

## 2 ЗАДАНИЕ

1. Построить таблицу 1 (5x5). В качестве исходных данных записать свою фамилию.

2. Найти минимальное время транспортировки, используя «жадный» алгоритм. Написать программу, реализующую «жадный» алгоритм для произвольной матрицы (10x10).

3. Разработать программу для нахождения минимального и максимального времени транспортировки методом полного перебора.

4. Оценить, к какому значению ближе решение, найденное в п.2.

5. Оценить вычислительную сложность алгоритмов.

## 3 ТРЕБОВАНИЕ К ОТЧЕТУ

В отчете должны быть отображены следующие пункты:

1. Задание.

2. Краткие теоретические сведения.

3. Пример решения задания для своего варианта (только «жадный» алгоритм). 4. Схемы алгоритмов.

5. Листинги основных частей программы.

6. Тесты.

7. Результат выполнения программы.

## 4 КОНТРОЛЬНЫЕ ВОПРОСЫ

1. В чем заключается суть «жадных» алгоритмов?

2. В чем заключается суть алгоритмов полного перебора.

3. Область применения «жадных» алгоритмов.

4. Область применения алгоритмов полного перебора.

5. Оценить вычислительную сложность «жадного» алгоритма.

6. Оценить вычислительную сложность алгоритма полного перебора.

7. Какой из рассмотренных алгоритмов гарантирует нахождение абсолютного минимума?