



ИНТЕЛЛЕКТУАЛЬНЫЙ
МЕГАПОЛИС

МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ



ИТ-класс

В МОСКОВСКОЙ ШКОЛЕ

**НАПРАВЛЕНИЯ
РОБОТОТЕХНИКА, СОЗДАНИЕ
ЦИФРОВЫХ ДВОЙНИКОВ,
ИНФОРМАЦИОННАЯ
БЕЗОПАСНОСТЬ И ТЕХНОЛОГИИ
СВЯЗИ, БОЛЬШИЕ ДАННЫЕ И
ТЕХНОЛОГИИ ИСКУССТВЕННОГО
ИНТЕЛЛЕКТА**

ТЕОРЕТИЧЕСКИЙ ЭТАП

**МОСКВА
2025**



ИНТЕЛЛЕКТУАЛЬНЫЙ
МЕГАПОЛИС

МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ РАЗРАБОТАНЫ:

Абрамешин Андрей Евгеньевич, профессор ДЭИ МИЭМ НИУ ВШЭ

Бубнова Мария Андреевна, ассистент ДКИ МИЭМ НИУ ВШЭ

Коваленко Алексей Анатольевич, инженер НИУ ВШЭ

Пожидаев Евгений Димитриевич, профессор-исследователь ДЭИ МИЭМ НИУ ВШЭ

МОСКВА
2025

Теоретический этап Конкурса проводится в очном дистанционном формате с использованием технологии прокторинга. Участникам необходимо иметь компьютер (ПК или ноутбук) с выходом в Интернет, веб-камерой и микрофоном, а также смартфон (или планшет) со стабильным интернетом и приложением для считывания QR-кодов. Прохождение диагностики на мобильных устройствах невозможно. Требуется предварительная настройка оборудования. Браузер разрешается использовать только для просмотра заданий теоретического этапа, внесения ответов и прохождения процедуры прокторинга.

При выполнении работы обеспечивается строгое соблюдение порядка организации и проведения Конкурса.

Во время выполнения работы разрешается дополнительно использовать следующее оффлайн программное обеспечение, предварительно установленное на компьютер (ПК или ноутбук): непрограммируемый встроенный калькулятор, текстовый и графический редакторы, среды программирования. Использование иного программного обеспечения ведёт к отклонению результатов написания работы.

На выполнение заданий теоретического этапа Конкурса отводится **120 минут**. Во время проведения мероприятия участник может выйти из зоны проведения мероприятия не более чем на 5 минут, предупредив проктора на камеру. Мероприятие не продлевается на время отсутствия участника.

Конкурсные задания разработаны преподавателями организаций высшего образования, участвующих в реализации проекта «ИТ-класс в московской школе».

Индивидуальный вариант участника формируется автоматизировано во время проведения теоретического этапа из базы проверочных заданий.

В работе используются задания с выбором ответа и кратким ответом.

Задание считается выбранным, если на него дан ответ. Участник может изменить свой выбор в процессе выполнения работы путём удаления ответа к заданию и сохранения нового ответа.

Задания 5 и 6 предусматривают решение одного задания из каждой пары заданий на выбор участника.

№	Тип задания*	Предмет	Проверяемые КЭС за 10-11 класс	Уровень**	Контролируемые требования к предъявляемым умениям
1	ВО	Математика	Элементы теории графов	Б	Знать: – основные понятия теории графов (вершина, ребро, степень, ориентированность); – структуру и свойства деревьев, в том числе двоичных; – понятие связности и компоненты связности; – определение пути, Эйлера и Гамильтонова путей. Уметь: – применять базовые понятия графов при решении задач; – строить и анализировать двоичные деревья; – определять компоненты связности в графе; – находить и проверять наличие Эйлера и Гамильтонова путей
2	ВО	Математика	Арифметический	Б	Знать:

			корень n-ой степени		– форму и свойства уравнений с одной переменной; – методы решения линейных и квадратных уравнений; – принципы решения систем линейных и квадратных уравнений. Уметь: – решать уравнения с одной переменной; – применять методы для нахождения решений линейных и квадратных уравнений; – составлять и решать системы уравнений
3	ВО	Математика	Показательные уравнения	Б	Знать: – свойства показательной функции; – формы простейших показательных уравнений и неравенств; – методы решения показательных уравнений. Уметь: – преобразовывать и решать показательные уравнения; – решать простейшие показательные неравенства; – применять свойства показательных выражений при упрощении
4	КО	Информатика	Информационная безопасность	У	Знать: – основы информационной безопасности и защиты данных; – методы предотвращения несанкционированного доступа; – назначение резервного копирования и шифрования; – типы вредоносного ПО и способы борьбы с ними. Уметь: – использовать базовые методы защиты информации на устройствах; – выполнять резервное копирование данных; – применять шифрование для защиты конфиденциальной информации – использовать антивирусные программы для обеспечения безопасности
5	КО	Информатика	Основы алгебры логики	У	Знать: – основные логические операции: импликация, эквиваленция, исключающее «или»; – структуру логического выражения и таблицы истинности; – методы решения логических уравнений и систем. Уметь: – выполнять логические преобразования с использованием указанных операций;

					– восстанавливать логические выражения по таблице истинности; – решать логические уравнения и системы логических уравнений
6	КО	Информатика	Введение в программирование	У	Знать: – основы формализации алгоритма, принципы работы машины Тьюринга; – типы данных: целочисленные, вещественные, строковые, файловые; – принципы работы с массивами, стеками, очередями, списками, деревьями, графами; – основные алгоритмы сортировки и поиска; – методы оценки вычислительной сложности. Уметь: – составлять и реализовывать алгоритмы обработки чисел, строк, файлов; – решать задачи с использованием массивов, стеков, очередей и деревьев; – реализовывать алгоритмы поиска путей в графе, включая алгоритм Дейкстры; – применять сортировки (пузырьком, выбором, вставками, слиянием) на практике; – оценивать эффективность алгоритма по времени выполнения и сложности

Решение кейса из демоварианта (Вариант 1)

Задание 1

Садовник организовал дорожки в саду между беседками в виде графа – дерева. Сколько дорожек в саду, если известно, что беседок 10 штук?

Выберите один верный вариант ответа:

- А) 8;
- Б) 9;
- В) 10;
- Г) 11.

Решение:

В дереве с n вершинами всегда ровно $n-1$ ребро. Если беседок 10, то число дорожек $= 10 - 1 = 9$.

Ответ: Б

Задание 2

Решите уравнение $2x^4 - 3x^2 + 1 = 0$. Выберите из предложенных вариантов ответа значение перемножения корней без учёта наименьшего.

- А) -0,5;
- Б) 0,5;
- В) 1;
- Г) -1.

Решение:

Пусть $x^2 = t$, тогда уравнение примет вид: $2t^2 - 3t + 1 = 0$.

Решим квадратное уравнение:

$$D = (-3)^2 - 4 \cdot 2 \cdot 1 = 9 - 8 = 1$$

$$t_{1,2} = (3 \pm \sqrt{1}) / (2 \cdot 2) = (3 \pm 1) / 4$$

$$t_1 = 1, t_2 = \frac{1}{2}$$

$$\text{Из } x^2 = 1 \rightarrow x = \pm 1$$

$$\text{Из } x^2 = \frac{1}{2} \rightarrow x = \pm 1/\sqrt{2}$$

Итак, корни уравнения: $x = 1, -1, 1/\sqrt{2}, -1/\sqrt{2}$

Наименьший корень: -1

Остальные корни: $-1/\sqrt{2}, 1/\sqrt{2}, 1$

Произведение этих трёх корней: $1 \times (1/\sqrt{2}) \times (-1/\sqrt{2}) = -1/2$

Таким образом, ответ $-1/2$

Ответ: А

Задание 3

Решите уравнение $2^{x-1} = 8$. Возведите найденный корень в третью степень.

Выберите один верный вариант ответа:

- А) 27;
- Б) 8;
- В) 64;
- Г) 1.

Решение:

Показательное уравнение: $2^{(x-1)} = 8$

$$8 = 2^3$$

$$2^{(x-1)} = 2^3$$

$$x - 1 = 3$$

$$x = 4$$

Возводим x в третью степень: $4^3 = 64$

Ответ: В

Задание 4

Настя хочет зашифровать строку "cat" в виде её хэша SHA-256. Какой хеш будет получен после применения библиотеки *hashlib* языка программирования Python?

Решение:

Используем библиотеку *hashlib* языка Python для вычисления SHA-256 хэша строки 'cat':

Код:

```
1 import hashlib
2 print(hashlib.sha256('cat'.encode()).hexdigest())
3
```

77af778b51abd4a3c51c5ddd97204a9c3ae614ebccb75a606c3b6865aed6744e

Ответ: 77af778b51abd4a3c51c5ddd97204a9c3ae614ebccb75a606c3b6865aed6744e

Задание 5

Напишите программу на языке Python, которая перебирает все возможные комбинации значений двух логических переменных a и b , принимающих значения 0 или 1. Для каждой комбинации вычислите значение выражения: $(a \text{ or } b) \text{ and not } (a \text{ and } b)$. Подсчитайте, в скольких комбинациях это выражение принимает значение «истина» (*true*). Допускается использование только стандартных логических операций. Выведите на экран одно целое число – количество таких комбинаций. Внесите его в поле ответа.

Решение:

Программа перебирает все значения a и b из $\{0, 1\}$ и вычисляет $(a \text{ or } b) \text{ and not } (a \text{ and } b)$.

Это выражение — реализация операции исключающего ИЛИ (XOR).

Подсчитываем, в скольких комбинациях результат — True:

Код:

```
1 count = 0
2 for a in [0, 1]:
3     for b in [0, 1]:
4         if (a or b) and not (a and b):
5             count += 1
6 print(count)
7
```

2

Ответ: 2

Задание 6

Напишите программу на языке Python, которая перемножает все простые числа от 2 до 19 включительно. Используйте вложенные циклы и проверку делимости. Запрещается использовать встроенные функции и внешние модули. Программа должна состоять не более чем из 15 строк. В поле ответа необходимо внести результат перемножения простых чисел, полученный в результате работы написанной программы.

Решение:

Задача — перемножить все простые числа от 2 до 19.

Код с вложенными циклами и проверкой делимости:

```
1 res = 1
2 for i in range(2, 20):
3     is_prime = True
4     for j in range(2, int(i ** 0.5) + 1):
5         if i % j == 0:
6             is_prime = False
7             break
8     if is_prime:
9         res *= i
10 print(res)
11
```

9699690

Ответ: 9699690

Подробные решения заданий 1–6 (Вариант 2)

Задание 1

Если в графе существует Эйлеров путь, то какое максимальное количество вершин нечётной степени может быть в графе?

Выберите один верный вариант ответа:

- А) 0;
- Б) 1;
- В) 2;
- Г) 3.

Решение:

Эйлеров путь — это путь, проходящий по каждому ребру ровно один раз.

Эйлеров путь существует тогда и только тогда, когда в графе:

- ровно 2 вершины нечётной степени (начальная и конечная точки пути),
- все вершины имеют чётную степень (в этом случае существует и Эйлеров цикл).

Ответ: В

Задание 2

Решите уравнение $x^4 + 5x^2 - 6 = 0$. Выберите сумму корней уравнения из предложенных вариантов ответа.

- А) 1;
- Б) -1;
- В) 0;
- Г) 2.

Решение:

Пусть $x^2 = t$. Тогда уравнение примет вид: $t^2 + 5t - 6 = 0$.

Найдём дискриминант:

$$D = 5^2 + 4 \cdot 6 = 25 + 24 = 49$$

$$t_{1,2} = (-5 \pm \sqrt{49}) / 2 = (-5 \pm 7) / 2$$

$$t_1 = 1, t_2 = -6$$

$$t = 1 \Rightarrow x^2 = 1 \Rightarrow x = \pm 1$$

$t = -6 \Rightarrow$ корней нет, так как $x^2 = -6$ не имеет действительных решений.

$$\text{Сумма корней: } 1 + (-1) = 0$$

Ответ: В

Задание 3

Решите уравнение $4^{x-1} = 64$. Извлеките квадратный корень из найденного значения корня уравнения.

Выберите один верный вариант ответа:

- А) 1;
- Б) 3;
- В) 2;

Г) 4.

Решение:

Показательное уравнение: $4^{(x-1)} = 64$

$$64 = 4^3$$

$$4^{(x-1)} = 4^3$$

$$x - 1 = 3$$

$$x = 4$$

Извлечь квадратный корень из x : $\sqrt{4} = 2$

Ответ: В

Задание 4

Сообщение «1010» зашифровали с помощью ключа «1100» операцией XOR. Какой результат будет получен?

Решение:

Операция XOR побитово складывает 2 строки:

Сообщение: 1010

Ключ: 1100

Результат:

$$1 \oplus 1 = 0$$

$$0 \oplus 1 = 1$$

$$1 \oplus 0 = 1$$

$$0 \oplus 0 = 0$$

Итог: 0110

Ответ: 0110

Задание 5

Напишите программу на языке Python, которая перебирает все комбинации значений x и y из множества $\{0, 1\}$ и проверяет, при каких парах переменных выполняется логическое выражение: $(x \text{ or } y) \text{ and } (\text{not } x \text{ or } y) == 0$. Выведите все такие пары (x, y) на экран. Используйте стандартные логические операторы Python. В поле ответа необходимо ввести количество пар, которое получилось при выводе в результате работы написанной программы.

Решение:

```
1 count = 0
2
3 for x in [0, 1]:
4     for y in [0, 1]:
5         if ((x or y) and (not x or y)) == 0:
6             print(f"x = {x}, y = {y}")
7             count += 1
8
9 print("Количество пар:", count)
```

```
x = 0, y = 0
x = 1, y = 0
Количество пар: 2
```

Ответ: 2

Задание 6

Напишите программу на языке Python: создайте пустой стек, последовательно добавьте в стек числа: 1, затем 2, затем 3. Извлеките все элементы из стека и выведите их на экран. Используйте только список `list`

и методы `.append()` и `.pop()`. Не используйте библиотеки и готовые реализации стека. В поле ответа введите число, которое будет получено при выводе, если не использовать перенос на другую строку, пробелы и посторонние символы.

Решение:

```
1 stack = []
2
3 # Добавляем элементы
4 stack.append(1)
5 stack.append(2)
6 stack.append(3)
7
8 # Извлекаем и выводим элементы
9 print(stack.pop(), end="")
10 print(stack.pop(), end="")
11 print(stack.pop(), end="")
12
```

321

Ответ:321

Подробные решения заданий 1–6 (Вариант 3)

Задание 1

Вычислите количество рёбер в полном графе с пятью вершинами.

Выберите один верный вариант ответа:

- А) 10;
- Б) 11;
- В) 12;
- Г) 13.

Решение:

В полном графе с n вершинами число рёбер вычисляется по формуле: $n(n-1)/2$

Если вершин 10, то:

$$5 \times (5 - 1) / 2 = 10$$

Ответ: А

Задание 2

Найдите корни уравнения $x^4 - 5x^2 + 4 = 0$. Выберите из предложенных вариантов ответа результат произведения всех корней уравнения без учёта наибольшего.

- А) 4;
- Б) -4;
- В) 2;
- Г) 1.

Решение:

Пусть $x^2 = t$. Тогда уравнение примет вид: $t^2 - 5t + 4 = 0$.

Найдём дискриминант:

$$D = (-5)^2 - 4 \cdot 1 \cdot 4 = 25 - 16 = 9$$

$$t_{1,2} = (5 \pm \sqrt{9}) / 2 = (5 \pm 3) / 2$$

$$t_1 = 4, t_2 = 1$$

$$t = 4 \Rightarrow x^2 = 4 \Rightarrow x = \pm 2$$

$$t = 1 \Rightarrow x^2 = 1 \Rightarrow x = \pm 1$$

Следовательно, корни уравнения: $x = -2, -1, 1, 2$

Наибольший корень: 2

Остальные корни: $-2, -1, 1$

$$\text{Произведение: } (-2) \times (-1) \times 1 = 2$$

Ответ: В

Задание 3

Решите неравенство $2^{x+1} > 8$.

Выберите один верный вариант ответа:

- А) $x > 1$;
- Б) $x > 2$;
- В) $x < 2$;
- Г) $x < 3$.

Решение:

Представим 8 как степень двойки: $8 = 2^3$

Если основания одинаковые и больше 1, то сравниваем показатели:

$$x + 1 > 3$$

$$x > 3 - 1 \Rightarrow x > 2$$

$$x > 2$$

Ответ: Б

Задание 4

Закодированное слово «ЙМБ» было получено после применения шифра Цезаря со сдвигом на 2 влево. Определите исходное слово, запишите его заглавными буквами в поле ответа.

Решение:

Шифр Цезаря — это замена каждой буквы на букву, сдвинутую в алфавите на заданное количество позиций. Классический вариант — это сдвиг на три позиции.

Сдвиг на 2 влево означает, что каждая буква исходного слова была сдвинута на 2 буквы назад по алфавиту, чтобы получить «ЙМБ». Чтобы расшифровать, нужно сдвинуть каждую букву на 2 буквы вперёд.

Й → Л (на 2 буквы вперёд)

М → О

Б → Г

Таким образом, исходное слово: ЛОГ

Ответ: ЛОГ

Задание 5

Напишите программу на языке Python, которая перебирает все возможные комбинации значений логических переменных x и y (0 и 1) и находит те случаи, когда выражение $x == (not y)$ истинно. Программа должна выводить на экран такие пары (x, y) . Используйте только стандартные логические операции и конструкции Python. В решении не допускается подключение библиотек. В поле ответа необходимо ввести количество пар, которое было получено при выводе в результате работы написанной программы.

Решение:

```
1 count = 0
2
3 for x in [0, 1]:
4     for y in [0, 1]:
5         if x == (not y):
6             print(f"x = {x}, y = {y}")
7             count += 1
8
9 print("Количество пар:", count)
10
```

```
x = 0, y = 1
x = 1, y = 0
Количество пар: 2
```

Ответ: 2

Задание 6

Реализуйте алгоритм пузырьковой сортировки для списка из n случайных чисел, числа уже заданы в коде программы. Необходимо отсортировать список по возрастанию с помощью пузырьковой сортировки, вывести отсортированный список на экран, подсчитать и вывести на новой строке общее число сравнений, сделанных алгоритмом.

Запрещено использовать встроенные функции сортировки и модули. В поле ответа необходимо ввести только вычисленное число сравнений, полученное в результате работы программы для списка $arr = [5, 1, 10, 3, 4]$.

Решение:

Алгоритм пузырьковой сортировки сортирует список и подсчитывает количество сравнений.

Пример: $arr = [5, 1, 10, 3, 4]$

Код:

```
1 arr = [5, 1, 10, 3, 4]
2 n = len(arr)
3 count = 0
4 for i in range(n):
5     for j in range(0, n-i-1):
6         count += 1
7         if arr[j] > arr[j+1]:
8             arr[j], arr[j+1] = arr[j+1], arr[j]
9 print(count)
10
```

10

Ответ: 10

Ключевые моменты:

Критерии оценивания: если задание решено верно, участник получает максимальный балл, иначе 0 баллов.

Описание хода практической части в случае очной дистанционной формы проведения этапа Конкурса: *категорически нельзя (ведет к отклонению работы): веб-поиском, методическими рекомендациями по направлениям Конкурса*. Во время проведения мероприятия участник может выйти из зоны проведения мероприятия не более чем на 5 минут, предупредив *проктора на камеру*. Мероприятие не продлевается на время отсутствия участника.

Трудности при подготовке – во избежание трудностей рекомендуется ознакомиться с полным комплектом методических материалов, самостоятельно решить демовариант, сверяясь с ответами и ходом решения, предложенном в методических материалах.

Задание 1: важные понятия.

Задание 1 во всех вариантах связано с базовыми понятиями теории графов, в том числе с деревьями, полными графами, степенями вершин и Эйлеровыми путями.

Граф — это множество вершин, соединённых рёбрами. Граф может быть:

- ориентированным (стрелки, важен порядок)
- неориентированным (обычные линии между вершинами)

Степень вершины — это количество ребер, инцидентных этой вершине. В ориентированных графах различают степень входа (количество входящих ребер) и степень выхода (количество исходящих ребер)

Пример: если вершина соединена с тремя другими, её степень — 3.

Сумма всех степеней вершин всегда равна удвоенному количеству рёбер.

Дерево — это связный граф без циклов.

Важное свойство дерева: если в дереве n вершин, то количество рёбер всегда $n - 1$.

Полный граф — простой неориентированный граф, в котором каждая пара различных вершин смежна.

Полный ориентированный граф — ориентированный граф, в котором каждая пара различных вершин соединена парой ребер с противоположными ориентациями.

Формула для числа рёбер в полном графе с n вершинами: $n(n - 1) / 2$

Эйлеров путь — это путь, проходящий через все его ребра.

Условия существования Эйлерова пути в неориентированном графе:

- граф связный.
- в графе не более двух вершин с нечётной степенью.

Если таких вершин 0 — существует Эйлеров цикл (начинается и заканчивается в одной вершине).

Если ровно 2 — существует Эйлеров путь (между двумя вершинами нечётной степени).

Если больше 2 — ни путь, ни цикл невозможны.

Задание 2: ключевые моменты

Биквадратные уравнения — это уравнения вида: $x^4 + a \cdot x^2 + b = 0$.

Рекомендуется использовать замену: $y = x^2$.

Можно представить в виде: $x^4 + a \cdot x^2 + b = 0$.

Решается заменой переменной: $y = x^2 \rightarrow$ уравнение становится квадратным: $y^2 + a \cdot y + b = 0$.

После подстановки $y = x^2$, решается обычное квадратное уравнение.

Далее для каждого найденного y решается уравнение $x^2 = y$:

- Если $y > 0$, уравнение имеет два корня: $x = \pm\sqrt{y}$.
- Если $y = 0$, один корень: $x = 0$.
- Если $y < 0$, вещественных корней нет.

Подсказка:

- Не забывайте: после замены всегда нужно возвращаться к x , решая $x^2 = y$.
- При поиске произведения не перепутайте знаки: $(-a) \cdot (a) = -a^2$.
- При суммировании учитывайте, что противоположные числа дают 0 (например, -3 и 3).
- Обратите внимание на действия, которые нужно сделать с корнем уравнения.

Задание 3: важные моменты

Как решать показательные уравнения

1. Привести обе стороны уравнения к одному основанию (если это возможно).
2. При равных основаниях приравнять показатели.
3. Решить полученное линейное уравнение.
4. Выполнить дополнительные действия, если они указаны (например, возвести результат в степень или извлечь корень).

Подсказки к выполнению

- Преобразуйте числа к степеням известных оснований: $8 = 2^3$, $27 = 3^3$, $125 = 5^3$ и т.п.

Задание 4: Хэш-функции. Шифр Цезаря. XOR.

.

1. Хэш-функции (SHA-256)

Хэш-функции преобразуют произвольную строку в фиксированную последовательность символов. SHA-256 — криптографическая хэш-функция, генерирующая 64-значный шестнадцатеричный код. Используется для проверки целостности данных. Изменение одного символа в исходной строке полностью меняет хэш.

Пример (на Python):

```
import hashlib
hashlib.sha256('cat'.encode()).hexdigest()
```

2. Шифр Цезаря

Шифр Цезаря — простой способ шифрования, при котором каждая буква заменяется на букву, находящуюся на фиксированное число позиций раньше или позже в алфавите.

Важно учитывать порядок букв в русском алфавите и то, что шифрование может быть прямым или обратным.

3. Операция XOR

XOR (исключающее ИЛИ) — побитовая логическая операция:

$1 \text{ XOR } 1 = 0$

$0 \text{ XOR } 0 = 0$

$1 \text{ XOR } 0 = 1$

$0 \text{ XOR } 1 = 1$

Подсказки:

- Для хэширования всегда используйте `encode()` перед передачей строки в хэш-функцию.
- Для XOR убедитесь, что сообщение и ключ одной длины и представлены в виде битовых строк.
- При работе с шифром Цезаря желательно выписать алфавит или использовать таблицу символов для точного сдвига.

Задание 5

Логические переменные

Логические переменные могут принимать два значения: 0 (False) и 1 (True). В Python они участвуют в логических выражениях.

Комбинаций двух переменных a и b — четыре: (0,0), (0,1), (1,0), (1,1).

Основные логические операторы

- $\text{not } x$ — логическое отрицание (инвертирует значение x).
- $x \text{ and } y$ — логическое И (True, если оба значения True).
- $x \text{ or } y$ — логическое ИЛИ (True, если хотя бы одно значение True).
- $x == y$ — проверка равенства значений.

Подсказки:

- Не путайте логическое И (and) с побитовым (&), и логическое ИЛИ (or) с побитовым (|).
- Всегда проверяйте, что означает вопрос: сколько раз выражение ложно или сколько раз — истинно?
- Лучше делать таблицу значений или запускать короткую программу для перебора комбинаций.

Задание 6

Задание 6 предлагает участникам Конкурса решить либо задачи на сортировки, либо на другие алгоритмы, которые изучаются на уроках информатики.

Пузырьковая сортировка

Пузырьковая сортировка — простой алгоритм, который многократно проходит по списку, сравнивает пары соседних элементов и меняет их местами, если они идут в неправильном порядке. Каждый проход «всплывает» самый большой элемент в конец списка.

Псевдокод алгоритма

Для массива длины n :

for i in range(n):

 for j in range(0, $n - i - 1$):

 if $\text{arr}[j] > \text{arr}[j+1]$:

$\text{arr}[j], \text{arr}[j+1] = \text{arr}[j+1], \text{arr}[j]$

Подсчёт числа сравнений

Каждая итерация внутреннего цикла делает одно сравнение ($\text{arr}[j] > \text{arr}[j+1]$).

Общее число сравнений для n элементов:

$$(n-1) + (n-2) + \dots + 1 = n(n-1)/2$$

Подсчёт присваиваний

Каждая операция обмена местами включает три присваивания.

То есть на каждую перестановку — 3 присваивания.

Общее число зависит от исходного порядка элементов (в отсортированном массиве перестановок не будет).