

Методические рекомендации по подготовке учащихся к Единому государственному экзамену 2016 года (информатика)

*Бреслав Р.Б., преподаватель кафедры
инновационных образовательных технологий СПб АППО*

Анализ результатов ЕГЭ–2015

В 2015 году Петербургские школьники традиционно продемонстрировали высокие результаты по информатике и ИКТ. Однако, по сравнению с 2014 годом средний балл за экзамен, как всех выпускников вообще, так и выпускников текущего года стал ниже, а количество результатов ниже минимального порога возросло практически в 2 раза.

Такое изменение связано, в первую очередь, с изменениями КИМ-2015 по сравнению с предыдущим годом — количество заданий экзамена сократилось за счёт исключения наиболее простых из них.

Анализ диаграммы первичных баллов позволяет предположить о двух существенно различающихся уровнях подготовленности учащихся, пришедших на экзамен, что также носит объективный характер.

Стоит так же отметить, что в 1,5 раза выросло количество учащихся, получивших высокий балл (81 и выше).

Общее количество выпускников, выбирающих экзамен, практически не меняется, однако есть тенденция к увеличению числа выпускников прошлых лет и снижению числа выпускников СПО (в 2015 году выпускников СПО на экзамене не было).

КИМ-2015 состоял из 23 задач тестовой части и 4 задач части с развернутым ответом, традиционно проверяемой экспертами. По сравнению с КИМ-2014 в варианте количество заданий базового уровня уменьшилось на 3, повышенного – на 2. Количество задач с выбором ответа сокращено с 13 до 3, в связи с чем результаты сравнения выполнения отдельных заданий ЕГЭ-2014 и ЕГЭ-2015 могут быть не совсем корректны. Тем не менее,

настораживает то, что результаты по 5 заданиям базового (1, 6, 9, 10, 12) и 3 заданиям повышенного (14, 18, 22) уровня заметно ухудшились. В первую очередь это связано с новизной формулировки заданий, то есть, фактически, задания были сформулированы в виде, сильно отличающимся от демо-версии экзамена, пробных вариантов, сборников задач. Например, в задаче 10 («знания о методах измерения количества информации»), требовалось найти объём графического файла по заданным параметрам. В такой формулировке задача уже встречалась ранее, однако, последние 3 года демо-версии экзамена содержат задачи на вычисление объёма звукового файла, что могло помешать выпускникам, готовым лишь к задачам демо-версии, справиться с таким заданием.

Так же «проблемной» можно назвать задачу 11 (базовый уровень, «рекурсивный алгоритм»). Однако результаты намного лучше, чем в предыдущем году, когда задание было существенно переработано (по сравнению с демо-версией).

Уменьшение общего числа заданий по отношению к 2014 году при сохранении общего времени продолжительности экзамена, не сделало экзамен проще в том числе за счёт увеличения вычислительной составляющей. Скорее всего, тенденция к дальнейшему увеличению этой составляющей сохранится.

Улучшились результаты по заданиям 17 («умение осуществлять поиск информации в Интернете»), 20 («анализ алгоритма, содержащего вспомогательные алгоритмы, цикл и ветвление»), 21 («умение анализировать программу, использующую процедуры и функции») и 23 («система логических уравнений»).

Результаты выполнения заданий части 2 в целом выше, чем в предыдущие годы, за исключением задания 26 (С3). Следует отметить, что задания 24 (С1) и 25 (С2) не сильно отличались от демо-версии, а задание 27 (С4) является небольшим, но естественным усложнением задания демо-версии.

Снижение результатов по задаче 26 объясняется не только незначительным отличием от демо-версии (игра стала двумерной), но и ужесточением критериев проверки – в каждом из заданий выпускникам предстояло определить выигрывающего игрока, строго описать стратегию, привести доказательство правильности стратегии и определить количество ходов, приводящих к победе (правда, отсутствие последней части не приводило к снижению баллов, в случае выполнения первых трёх частей).

Одной из особенностей ЕГЭ-2015 стало опубликование на сайте ФИПИ КИМ досрочного этапа до проведения основного этапа. В отличие от предыдущего года, когда досрочный КИМ так же был опубликован, в 2015 году досрочный КИМ давал представление об изменениях, которые ожидали выпускников в основной период.

К сожалению, появившийся за несколько дней до экзамена в сети Интернет разбор задачи 27, оказал медвежью услугу выпускникам, поскольку незначительное изменение условия (проверка чётности произведения в основном варианте, против проверки нечётности в досрочном), привело к существенному изменению алгоритма решения, на что часть выпускников не обратило внимания.

Изменения в КИМ-2016

Единственным существенным изменением в КИМ-2016 стал отказ от заданий с выбором правильного ответа в тестовой части (задания 1–3). Данные задания будут переформулированы в задания с кратким ответом, в результате чего изменился порядок следования заданий в связи с изменившейся сложностью. В остальном структура работы по темам работы и количеству задач различных уровней сложности (базовый, повышенный, высокий) не изменилась.

Методические рекомендации

Основываясь на анализе ЕГЭ-2015 можно выделить следующие 3 проблемы, которые важным образом влияют на результаты выпускников:

- Шаблонность мышления

- Увеличение технической сложности КИМ
- Слабое владение математическим аппаратом, в том числе применительно к предмету

Шаблонность мышления заключается в том, что выпускники ожидают увидеть на экзамене задания аналогичные демо-версии, подготовительных вариантов с Интернет-ресурсов или литературы по подготовке к ЕГЭ. К сожалению, некоторые выпускники при подготовке ограничиваются только вариантом демо-версии текущего года.

Для уменьшения влияния шаблонности мышления при подготовке к ЕГЭ рекомендуется использовать не только задачи текущего года, но и архивы задач и демо-версий прошлых лет. Полезным мог бы оказаться открытый банк заданий ЕГЭ, однако, в том виде, в котором он представлен на сайте ФИПИ им пользоваться практически невозможно, в виду очень плохой структурированности материала (все задания банка разбиты всего на три больших группы).

Можно порекомендовать задания с сайта Решуегэ.рф (помимо банка задач и тренировочных вариантов сайт так же содержит архив диагностических работ) и архивы раздела ЕГЭ сайта К.Полякова.

Стоит обращать внимание на различные способы решения одной и той же задачи, даже если такой способ решения не является допустимым для формата ЕГЭ. Например, решение заданий с использованием электронных таблиц или написание простых программ. Такой подход в первую очередь применим к задачам раздела «Математические основы информатики» и чуть меньше к разделу «Теоретические основы информационно-коммуникационных технологий».

Увеличение технической сложности КИМ в первую очередь направлено на отсеивание нерациональных решений, например, использование полного перебора вариантов. Зачастую, нерациональность решения связана с вычислениями. Например, вычисления проводятся в десятичной системе счисления, как в промежуточной, в то время как те же действия, проведённые

в системе счисления с более удобным основанием, занимают на порядок меньше времени. В задачах на вычисление объёма памяти, учащиеся также, зачастую предпочитают проводить действия «в лоб», вместо того, чтобы заметить, что вычисляемое выражение сократимо, как минимум на степени 2. К сожалению, некоторые задания (например, второй вариант задания 4 по демо-версии), усложняясь, просто требуют большего объёма работы, не неся в этом никакой дополнительной идеи.

Проблема слабого владения математическим аппаратом особенно проявляется в следующих темах:

- Системы счисления
- Комбинаторика
- Теория чисел

Остановимся на каждой из них подробнее.

Системы счисления.

В условиях реализации ФГОС данная тема впервые изучается в курсе математики. Основными алгоритмами, которыми должны овладеть учащиеся, безусловно, являются алгоритмы перевода из десятичной системы счисления в недесятичную и обратно. Не менее важными являются алгоритмы сложения, вычитания в недесятичных системах. Для повторения и закрепления навыков работы с недесятичными системами счисления можно использовать электронные таблицы, в которых учащиеся могут реализовывать основные алгоритмы. Также рекомендуется вспомнить про алгоритмы перевода между системами счисления, при изучении темы «Циклы» раздела «Алгоритмы и программирование». При этом желательно, чтобы учащиеся не только могли бы самостоятельно написать цикл перевода числа из одной системы счисления в другую, но и узнавать эти алгоритмы в фрагментах программ.

Также, учащимся следует освоить некоторые навыки рациональных вычислений: алгоритм перевода между системами счисления, основания

которых являются степенями одного и того же числа; перевод числа, «почти равного» степени основания.

Пример 1 (демо-версия КИМ-2016. Задание 1).

Сколько единиц в двоичной записи шестнадцатеричного числа $12F0_{16}$?

Решение. Каждой цифре в 16-ичной записи числа соответствует 4 цифры двоичной записи. $1 = 0001$, $2 = 0010$, $F = 1111$, $0 = 0000$. Итоговое количество единиц – 6.

Варианты:

- посчитать количество нулей (возможные ошибки – не учитываются ведущие нули для цифр 2 и 0, либо излишне учитываются ведущие нули для цифры 1)
- посчитать количество единиц в восьмеричной записи, количество троек в четверичной записи (использование двоичной системы в качестве промежуточной)

Пример 2 (демо-версия КИМ-2016. Задание 16).

Значение арифметического выражения $9^8 + 3^5 - 9$ записали в системе счисления с основанием 3. Сколько цифр «2» содержится в этой записи?

Решение. Заметим, что $9^8 = 3^{16}$. В троичной системе счисления это число записывается $10\dots0_3$ (16 нулей). Осталось прибавить $3^5 = 100000_3$ и вычесть 100_3 . $100000_3 - 100_3 = 22200_3$. $10\dots0_3 + 22200_3 = 10\dots022200_3$ (между 1 и 2 – 11 нулей). Ответ: 3.

Замечание. Ни ход решения, ни ответ не меняются при замене 3 на произвольное натуральное число $a > 2$, 9 – на a^2 , а 2 – на $a - 1$. Например:

Значение арифметического выражения $256^8 + 16^5 - 256$ записали в системе счисления с основанием 16. Сколько цифр «F» содержится в этой записи?

Заметим, что аналогичная задача в двоичной системе счисления имеет «подводный камень» – необходимо не забыть старшую единицу.

Варианты:

- посчитать количество нулей в записи
- посчитать сумму цифр

В отличие от предыдущих тем, в курсе математики не рассматриваются поразрядные логические и битовые операции над двоичными числами. Тем не менее знание данных операций необходимо для понимания работы АЛУ процессора, адресации в TCP/IP сетях и пр.

Комбинаторика.

Данная тема так же частично проходит в курсе математики младших классов и проверяется в рамках ОГЭ. В идеале, из курса математики учащиеся должны знать правила сложения и умножения и уметь применять их на практике, при решении простых задач.

Помимо правил сложения и умножения учащимся могут быть полезны следующие знания:

- Подсчёт, нумерация, перечисление комбинаторных объектов
- Поиск оптимальных путей в графе (алгоритм Дейкстры)
- Подсчёт путей в графе (динамическое программирование)

Пример 1 (демо-версия КИМ-2016. Задание 10). Подсчёт количества комбинаторных объектов.

Игорь составляет таблицу кодовых слов для передачи сообщений, каждому сообщению соответствует своё кодовое слово. В качестве кодовых слов Игорь использует 5-буквенные слова, в которых есть только буквы П, И, Р, причём буква П появляется ровно 1 раз. Каждая из других допустимых букв может встречаться в кодовом слове любое количество раз или не встречаться совсем. Сколько различных кодовых слов может использовать Игорь?

Решение. В данном случае воспользуемся правилом умножения. Выберем место в кодовом слове, на котором будет стоять буква П (5 вариантов), затем заполним остальные 4 места. На каждое из них можно поставить любую из 2 оставшихся букв (4 раза по 2 варианта). Итого: $5 \cdot 2^4 = 80$ вариантов.

Заметно усложнить задачу можно, заменив простой подсчёт кодовых слов на их перечисление или нумерацию (например, в алфавитном порядке).

Слегка изменим условие задачи:

Игорь составляет таблицу кодовых слов для передачи сообщений, каждому сообщению соответствует своё кодовое слово. В качестве кодовых слов Игорь использует 5-буквенные слова, в которых есть только буквы К, О, Т, причём буква К появляется ровно 1 раз. Каждая из других допустимых букв может встречаться в кодовом слове любое количество раз или не встречаться совсем. Затем Игорь выписал все кодовые слова в алфавитном порядке и пронумеровал их. Вот начало этой таблицы:

1. КОООО
2. КОООТ
3. КООТО
4. КООТТ

Определите, под каким номером в таблице стоит слово КТТТО? Определите, какое слово стоит под номером 20? Какое слово стоит в таблице следующим после слова ТОКОТ?

Решение. Очевидно, что в составленной таблице сначала идут слова, начинающиеся на букву К. Оставшиеся буквы можно закодировать цифрами 0 и 1 ($O = 0$, $T = 1$) и рассматривать слово из оставшихся 4 букв как число, записанное в двоичной системе (что совпадёт с алфавитным порядком) от $OOOO = 0$ до $TTTT = 15$. Тогда слову КТТТО будет соответствовать число $TTTO = 14$. Такое слово, очевидно, стоит под номером 15. Затем в таблице пойдут слова, начинающиеся с ОК. Такие слова будут стоять с 17 по 24 строках таблицы и кодируются последними 3 буквами (от OOO до TTT). 20-й строке соответствует число 3 и окончание OTT . То есть слово в 20 строке – $OKOTT$. Слово $ТОКОТ$ стоит в блоке слов, начинающихся на $ТОК$ (от $ТОКОО$ до $ТОКТТ$) на 2 месте ($OT = 01$). Следующим за ним будет слово на 3 месте ($ТО$), то есть это слово $ТОКТО$.

Более сложные комбинаторные объекты (перестановки, сочетания, разбиения и т.д.) можно найти в задачах районных туров Всероссийской олимпиады школьников по программированию.

Пример 2 (демо-версия КИМ-2016. Задание 15). Подсчёт путей в графе.

1	4	2	1	5	16	7	28	28	28	56
---	---	---	---	---	----	---	----	----	----	----

Для решения данной задачи удобно использовать электронную таблицу. В ячейки таблицы заносятся формулы, соответствующие вычислению количеству способов попасть в данную вершину:

	A	B	C	D	E	F	G	H	I	J	K	L
1	A	B	B	Г	Д	Е	Ж	З	И	К	Л	М
2	1	=A2	=A2+B2+D2	=A2+E2	=A2	=B2+C2	...			=I2	=I2	=K2+J2

Варианты (данные усложнения в полной мере реализованы в следующем примере):

- Сколько существует различных путей из А в М, проходящих через Ж?
- Сколько существует различных путей из А М не проходящих через Г?

Пример 3. (демо-версия КИМ-2016. Задание 22). Подсчёт путей в графе.

Исполнитель Май15 преобразует число на экране.

У исполнителя есть две команды, которым присвоены номера:

1. Прибавить 1

2. Умножить на 2

Первая команда увеличивает число на экране на 1, вторая умножает его на 2.

Программа для исполнителя Май15 – это последовательность команд.

Сколько существует программ, для которых при исходном числе 2 результатом является число 29 и при этом траектория вычислений содержит число 14 и не содержит числа 25?

Траектория вычислений программы – это последовательность результатов выполнения всех команд программы. Например, для программы 121 при исходном числе 7 траектория будет состоять из чисел 8, 16, 17.

Решение (динамическое программирование). Несмотря на то, что формулировка задачи отличается от предыдущей, по сути это та же самая задача. Можно нарисовать граф, в котором число 2 будет соединено стрелками с 3 (соответствует первой команде) и 4 (соответствует второй

команде), число 3 – с 4 и 6 и т.д. Условие, накладываемое на траекторию означает то, что сначала надо из числа 2 получить число 14, а затем, из 14, число 29, при этом не проходя через число 25.

Решим первую половину задачи, составив таблицу:

2	3	4	5	6	7	8	9	10	11	12	13	14
1	1	2	2	3	3	5	5	7	7	10	10	13

3 можно получить только из 2 – 1 способ, 4 из 2 и 3 – $1+1=2$ способа, ... 14 из 13 и 7 – $10+3=13$ способов.

Вторую часть задачи можно решить, составив аналогичную таблицу, в которой будет пропущено число 25. Но можно заметить, что числа 26 и 27 получить из 14, не проходя через 25 невозможно. Следовательно, 28 можно получить только из 14 (13 способов получить из 2), а 29 – только из 28. Таким образом, получается, что число 29 при наложенных ограничениях на траекторию можно получить из 2 всего 13 способами.

Теория чисел в демо-версии КИМ-2016 присутствует только в задании 20, где требуется проанализировать алгоритм, частью которого является алгоритм Евклида вычисления НОД чисел. В КИМ-2015 теоретико-числовые идеи встречались в заданиях 14 (анализ алгоритма для исполнителя) и 18 (знание основных понятий и законов математической логики). Именно теоретико-числовая составляющая, вероятно, и стала наибольшей проблемой для выпускников при решении данных заданий.

НОД и НОК опять-таки изучаются в курсе математики в младших классах. Изучение данных тем, может не включать изучения алгоритма Евклида, свойств НОДа и НОКа, а так же их связей с множествами общих делителей и общих кратных. Так же учащиеся в старших классах, в основном, не помнят свойств делимости и не умеют применять их на практике. Для тренировки таких навыков рекомендуется обратить внимание на задачи 14 и 18 досрочного этапа ЕГЭ-2015 а так же задачи из архива с сайта К.Полякова за 2015 год.

Рекомендации по подготовке ко 2 части ЕГЭ.

Традиционно, при подготовке ко 2 части рекомендуется обратить внимание на критерии оценки заданий с развёрнутым ответом, которые представлены в демо-версии, поскольку они дают представление о требованиях, предъявляемых к правильному решению, а так же о возможных ошибках учащихся.

Задание 24 (С1). Умение прочесть фрагмент программы на языке программирования и исправить допущенные ошибки.

Задание традиционно состоит из нескольких пунктов, включающих в себя трассировку программы при заданных входных данных, определения случаев, когда программа работает верно, поиску и исправлению ошибок.

Не стоит заикливаться на формах задания, встречавшихся в КИМ предыдущих лет, но не стоит и игнорировать их.

В качестве подготовки можно использовать ошибочно работающие программы написанные учащимися при решении заданий 25 или 27. При этом выполняются следующие задания: «приведите пример данных, для которых программа работает верно», «приведите пример данных, для которых программа работает неверно», «исправьте все ошибки в предложенной программе».

Важное изменение последних лет в критериях задания 24 заключается в том, что при поиске и исправлении ошибок, верная строка, указанная в качестве ошибочной считается ошибкой и влечёт снижение баллов за задание. В первую очередь это касается мнимых синтаксических ошибок, например, отсутствие/наличие точки запятой в Паскале перед оператором `end`, отсутствие/наличие фигурных скобок в языках с C-подобным синтаксисом в условном операторе или цикле, тело которых состоит из одного оператора.

Указание двух таких псевдоошибочных строк даже если всё остальное решение абсолютно верно не позволяет поставить за задание 2 или более баллов!

Задание 25 (C2). Умения написать короткую (10–15 строк) простую программу на языке программирования или записать алгоритм на естественном языке.

Как и раньше, в демо-версии КИМ-2016, предлагается написать программу, обрабатывающую массив. Несмотря на то, что поставленная задача решается при помощи однопроходного алгоритма – цикл, в котором каждый элемент массива (или пара элементов, как в демо-версии) просматривается не более одного раза. Несмотря на то, что оптимальность написанной программы и используемого алгоритма в задании не учитывается, при подготовке следует обратить внимание именно на такие алгоритмы. Они не только проще в реализации, но и спокойно укладываются в предложенные ограничения по переменным.

Помимо оптимальности, в данной задаче так же не оцениваются синтаксические ошибки, не приводящие к искажению смысла программы.

Самыми популярными ошибками при выполнении задания 25 являются отсутствие инициализации переменной, выход за границы массива, неверно организованный цикл и отсутствие вывода ответа. Заметим, что ошибка выхода за границы массива характерна в первую очередь для задач, в которых рассматриваются пары соседних элементов. Остальные возможные ошибки перечислены в критериях к заданию 25.

Отметим, что поменялось отношение к решениям, использующим такие возможности современных языков программирования, как сортировка, фильтрация, поиск максимумов/минимумов. Если раньше такие решения считались допустимыми (что иногда позволяло записать решение задания в одну строчку), то теперь такие решения не засчитываются – предполагается, что выпускник должен реализовать алгоритм самостоятельно.

Алгоритм может быть записан на естественном языке, однако в таком случае он должен допускать «трансляцию» в традиционный язык программирования.

Задание 26 (С3). Умение построить дерево игры по заданному алгоритму и уметь обосновать стратегию.

Задание состоит из 3 частей, каждая из которых является очередным шагом решения общей задачи – полного анализа игры. Каждый следующий шаг опирается на предыдущий, в результате чего, ошибка на первом шаге может распространиться дальше. К сожалению, поскольку именно первый шаг является самым простым, именно на нём и появляются ошибки отсутствия описания стратегии или её доказательства. Одновременно описать стратегию и доказать правильность позволяет корректное дерево всех партий (представленное в виде дерева, таблицы или полного описания).

Типичные неверные решения первой части задания выглядят примерно так:

- При любом ходе Пети, Вася удваивает количество камней и выигрывает (описана стратегия, отсутствует доказательство)
- После первого хода Пети, количество камней становится от 40 до 72, в результате чего Вася получает 73 или более камня и выигрывает (присутствует доказательство, но отсутствует описание стратегии)

При подготовке заданий для подготовки, рекомендуется рассматривать эквивалентные игры, отличные по формулировкам, от присутствующих в демо-версиях и КИМ прошлых лет. Например, игра с 2 кучками камней, в которой разрешённые ходы «добавить 1 камень в любую кучку» и «увеличить количество камней в обеих кучках на 1» эквивалентна игре с фишкой на прямоугольной доске, где фишка может передвигаться на 1 клетку по горизонтали вправо, по вертикали вверх или по диагонали вправо-вверх.

Задание 27 (С4). Умение составлять собственные программы для решения задач средней сложности.

С 2015 года в задании предложено решить одну из 2 задач – А (написать программу, сохраняющую входные данные в массиве) и Б (написать оптимальную по памяти и/или времени программу). Задача А оценивается исходя из 2 баллов, задача Б – из 4. Экзаменуемый имеет право решить обе

задачи, при этом будет засчитан лучший результат. Поскольку задача А по сложности сопоставима с заданием 25 (С2), рекомендуется сначала записать решение задачи А, а потом приступить к задаче Б. Основное отличие задач заключается в том, что в задаче А речь идёт об обработке массива, в то время, как задача Б подразумевает обработку потоковых данных.

На примере задания демо-версии КИМ-2016 покажем, каким может быть подход учащегося при составлении алгоритма.

Переформулированное (для простоты понимания) задание выглядит так:

На вход программе подаётся число N , затем N положительных чисел не превосходящих 1000: a_1, a_2, \dots, a_N . Требуется найти минимальное чётное произведение $a_i a_j$ такое, что $i - j \geq 6$. Если такого произведения нет, то программа должна вывести -1 .

Поскольку мы имеем дело с потоковыми данными (то есть не можем сохранить все входные данные в памяти), программа должна уметь реагировать на появление нового числа из входного потока и в соответствии с ним обновлять результат, а так же все данные, необходимые для вычисления результата в тот момент, когда поступит следующее число из потока.

Определим, что необходимо для обновления результата. Если на вход поступило чётное число a_i , то минимальное чётное произведение, в которое оно будет входить равно $a_i \cdot \min(a_j \mid j \leq i-6)$. Если же a_i нечётное, то минимальное чётное произведение равно $a_i \cdot \min(a_j \mid a_j - \text{чётное}, j \leq i-6)$, при условии, что такой минимум существует. Пусть $\text{min}_6 = \min(a_j \mid j \leq i-6)$, а $\text{mine}_6 = \min(a_j \mid a_j - \text{чётное}, j \leq i-6)$ или -1 , если минимум не существует.

Тогда, алгоритм обновления результата (res) выглядит так (пусть a – число, которое только что было считано из потока):

```
if a mod 2 = 0 then newres := a*min6
  else newres := a*mine6;
if (newres > 0) and ((newres < res) or res=-1) then
  res := newres;
```

Теперь программа должна быть готова к приёму следующего числа из потока, то есть должны обновиться переменные minb и mineb .

Переменная minb может измениться, только если $a_{i-5} < \text{minb}$, в таком случае $\text{minb} = a_{i-5}$. Аналогично mineb может измениться только тогда, когда a_{i-5} чётно и $a_{i-5} < \text{mineb}$ или $\text{mineb} = -1$. Из чего делаем вывод, что нам надо хранить в памяти 5 значений $a_{i-5}, a_{i-4}, \dots, a_{i-1}$. Тогда после обновления результата, алгоритм должен взять из памяти число a_{i-5} , при необходимости обновить значения minb и mineb , и сохранить в памяти $a_{i-4}, a_{i-3}, \dots, a_i = a$.

Подобный подход применим ко всем задачам, обрабатывающим потоковые данные, которые можно решить без сохранения всего потока в памяти.

При объяснении данного подхода учащимся рекомендуется начинать с простых задач: нахождение минимума/максимума, суммы/произведения, вторых минимума/максимума и т.п. в потоке.

Рекомендуемые ресурсы.

1. ФИПИ. ДемOVERсии, кодификаторы, спецификации.
<http://fipi.ru/ege-i-gve-11/demoversii-specifikacii-kodifikatory>
2. ФИПИ. Открытый банк заданий ЕГЭ.
<http://www.fipi.ru/content/otkrytyy-bank-zadaniy-ege>
3. ФИПИ. КИМ-2015 досрочного этапа.
<http://www.fipi.ru/content/kim-ege-2015-dosrochnyy-period>
4. Официальный сайт ГИА в Санкт-Петербурге. <http://www.ege.spb.ru/>
5. ЕГЭ по информатике в Санкт-Петербурге.
<https://sites.google.com/site/appocio/ege>
6. Сайт К. Полякова. Подготовка к ЕГЭ по информатике.
<http://kpolyakov.spb.ru/school/ege.htm>
7. Решуегэ.рф. Информатика. <http://inf.reshuege.ru/>