

ПРЕДИСЛОВИЕ

Среди многочисленных конкурсов, в том числе по информатике, особое место занимает Международный конкурс «Бобер». Одним из его отличительных свойств является то, что базу задач, используемых в конкурсе, создают учителя и научные работники из более чем 65 стран-участников Конкурса. Эти задачи обсуждаются, важное значение в допуске задач к конкурсу имеет тот факт, сколько в каждой задаче содержится собственно информатики. Цель и назначение этих задач – представить в занимательной форме многие идеи информатики, которые изучаются в школе, такие как понятия алгоритма, упорядоченности, списков и их слияний, графов, деревьев и т.п.

Ввиду того, что проведение конкурса рассчитано на время, соответствующее одному уроку, на нем невозможно предлагать трудные и проблемные задачи, как это сделано, например, в Конкурсе по применению ИКТ в естественных науках, технологиях и математике «Конструируй, Исследуй, Оптимизируй». Кроме того, в конкурсе могут участвовать ребята разного уровня подготовки, не только так называемые «олимпиадники». Но именно поэтому задачи, представленные на конкурсе и имеющиеся в свободном доступе на сайте конкурса и в виде брошюр, могут использоваться на уроках информатики для индивидуальных и групповых заданий. Если учесть, что на конкурсе предлагаются задачи от совсем простых (3 балла) до достаточно сложных (9 баллов в 1–2 классе и 12 баллов в 4–11 классах), то всегда можно выбрать задания, соответствующие уровню обучаемого. Информатика стала одним из базовых предметов, представляющих основу школьного образования. Имеются школы, где информатику начинают преподавать с первого класса, во многих школах – со второго, в пятом – одиннадцатом классе информатика в том или ином виде преподается уже во всех школах.

Современные дети с малых лет спокойно управляют с компьютером, умеют найти нужную (а часто и совсем не нужную) информацию. Многие умеют использовать офисные программы. Но, используя информационные технологии, они должны понимать, где в них «зарыта» информатика и ее основные понятия.

Конкурс «Бобер» не ориентирован на проверку и оценку знаний учащихся в рамках программы школьного курса информатики и ИКТ и на подготовку к ЕГЭ. У него другая цель: с помощью занимательных за-

дач помочь обучаемому овладеть многими важными понятиями информатики.

Идея этого конкурса была предложена профессором Валентиной Дагене – заведующей кафедрой методики преподавания информатики института математики и информатики Вильнюсского университета, руководителем национальной олимпиады и автором многих книг по информатике. Название конкурса родилось во время её путешествия по Финляндии в 2003 году. «Будучи в Финляндии, – пишет В. Дагене, – я обратила внимание на деятельность бобров у берегов озер и рек. Бобр – упрямое животное, строящее запруды даже в быстротечных реках, долго работающее и достигающее своих целей. Кажется, будто бобры соревнуются – кто больше деревьев повалит и построит больше плотин» [1]. Так и получилось, что конкурс назвали в честь трудолюбивого, умного и живого бобра. Поскольку Литва является инициатором соревнования, сайт российского конкурса решено было называть BEBRAS – так «Бобёр» переводится на литовский язык (<http://bebras.ru>). Впервые конкурс проводился в Литве в 2004 году. Россия включилась в проведение конкурса в 2012 году. Организацию конкурса «Бобёр» в России осуществляет Оргкомитет конкурса «Конструируй, исследуй, оптимизируй» совместно с Факультетом компьютерных технологий и информатики Санкт-Петербургского государственного электротехнического университета «ЛЭТИ» им. В. И. Ульянова-Ленина. Конкурс проводится ежегодно в ноябре.

Каждая задача, несмотря на свою занимательную (для младших школьников – игровую) оболочку, всегда содержит некоторую информацию, относящуюся к понятиям, изучаемым в школе на уроках информатики, поэтому задачи, предлагавшиеся на конкурсе в разные годы, могут быть с большой пользой использованы в учебном процессе. Задание школьники смогут познакомиться с задачами, которые решают на уроках информатики и во внеурочное время их сверстники в других странах.

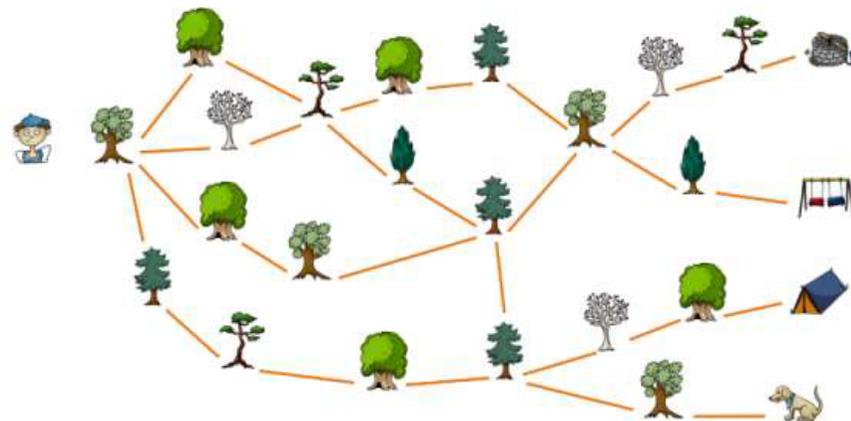
Задачи, предлагавшиеся на конкурсе в 2015 году, выпущены в виде отдельных брошюр.

Ниже всего лишь на примере части задач, предлагавшихся на конкурсе в 2013 году, показано, как и с какой целью их можно использовать на уроках информатики в начальной, основной и старшей школе. В задачах даны более подробные решения, чем на сайте, и более развернутые комментарии.

ПРИМЕРЫ ЗАДАЧ КОНКУРСА, КОТОРЫЕ МОГУТ БЫТЬ ИСПОЛЬЗОВАНЫ НА УРОКАХ ИНФОРМАТИКИ В МЛАДШИХ КЛАССАХ

Рассмотрим некоторые задачи, предлагавшиеся участникам 1–2 классов. Во время конкурса времени на раздумья отводилось мало, и в основном от участника требовалась внимательность и сообразительность. На уроке каждую задачу можно проанализировать, выяснить, чем руководствовался ученик, выбирающий тот или иной ответ, а главное, – обратить внимание на то, какая в каждой задаче прячется «информатика».

Задача 1. В лесу.  Словакия
Руперт пошел прогуляться в лес.



Он выбрал дорожку, в конце которой обнаружил собаку.

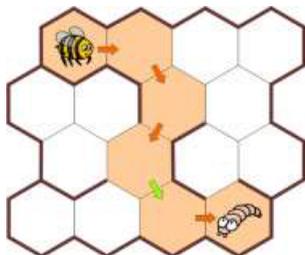
Мимо каких деревьев он прошел? Деревья представлены в том порядке, в котором Руперт видел их.

3. Можно ли попасть в левую клетку, используя заданное направление?

Ответ положительный. Переходим в верхнюю клетку и продолжаем анализировать возможные пути.

В результате будет выбран путь из 5 шагов, указанный на картинке:

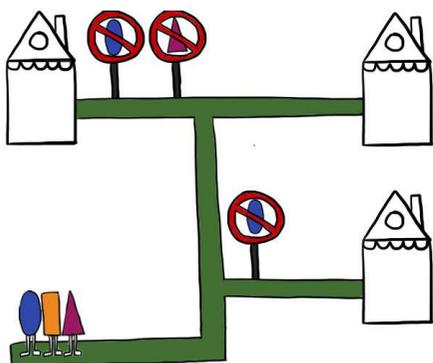
И опять в рассуждениях можно обращать внимание на логические операции: ИЛИ и НЕ при анализе каждого пути.



Мы построили алгоритм – последовательность команд для пчелы, которые описывают, как добраться до личинки за 5 шагов и при этом удовлетворить условию задачи: на предпоследнем шаге сделать нужный поворот. Подобные задачи часто решаются в различных областях информатики.

Для информатики знание законов логики очень важно. Для более подробного изучения логической операции отрицания НЕ можно выбрать, например, следующую задачу:

Задача 3. Дома. Словакия



Переместите геометрические фигурки в свои дома, учитывая запрещающие знаки около домов.

В верхний левый домик может быть поставлен НЕ синий овал, НЕ красный треугольник, так как эти знаки запрещены. Остается только оранжевый прямоугольник.

Из оставшихся двух фигурок в правый нижний домик можно поселить только красный треугольник, так как синий овал для него запрещен. Оставшийся синий овал помещаем в правый верхний домик.

В информатике, особенно в программировании, часто приходится повторять одни и те же действия, один и тот же алгоритм. Самый простой пример – строгое выполнение определенного режима дня в течение некоторого периода. Но также можно рассматривать повторяемое выполнение последовательности некоторых действий человеком или машиной. Компьютер для решения многих задач должен также выполнить последовательность элементарных команд. Такой алгоритм называют *циклическим* или просто *циклом*.

Если в задаче 1 обсуждалось понятие графа, то можно предложить ребятам подумать над более трудной задачей.

Задача 4. Мосты. Россия



Нужно расставить семь мостиков так, чтобы Василиса смогла прогуляться, пройдя по каждому мостику ровно один раз и вернуться обратно.

С решения этой задачи теория графов и начиналась.

Итак, имеется 3 острова и 2 берега. Если все острова соединить между собой и с берегами, то потребуется 8 мостов, а у нас их семь, следовательно, какой-то мост лишний.

Попробуем задать вопрос:

Сколько мостов должно соединять место, с которого Василиса отправится гулять по мостам, чтобы она смогла вернуться на то же место?

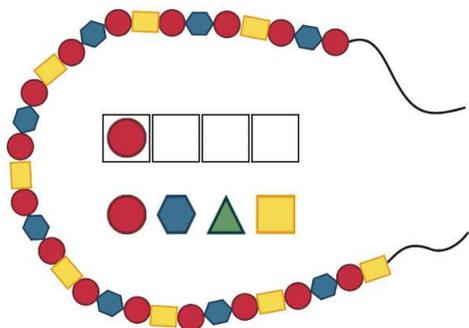
Очевидно, что их должно быть четное число (2 или 4).

На нижний берег и на маленький остров теперь ведут по 3 моста, следовательно, мост, их соединяющий, можно исключить. Теперь обход мостов можно начать с любого места.

Можно предложить ребятам проверить, что, начиная обход с любого места, можно в него вернуться.

Ну, а чтобы совсем было похоже на информатику, предложить ребятам нарисовать схематический чертеж: острова и берега изобразить точками – это вершины графа, соединить точки отрезками или дугами, изображающими мосты – это ребра графа, пронумеровать точки следующим образом: в точке 1 сходятся две дуги, в точках 2 и 3 – четыре дуги, в точках 4 и 5 – три дуги. Уберем одну дугу, соединяющую точки 4 и 5. Предложим расставить стрелочки, обозначающие направление обхода. Теперь можно разбить ребят на 5 групп и предложить каждой группе начать обход из точки с номером группы. Можно убедиться, что все обходы закончатся в точке, с которой обход начался.

Задача 5. Нанизывание бус. Словакия



Джейн хочет сделать новые бусы по образцу на рисунке. У нее есть машинка, которая нанизывает бусинки по заданному образцу. Какую последовательность бусинок нужно задать машинке, чтобы Джейн получила именно такие бусы?

В этой задаче последовательность бусинок повторяется по образцу, а то, что повторяется, называется *телом* цикла.

Преподавание информатики в начальной школе должно обеспечивать мотивацию информационной активности детей и пробуждение интереса к информатике и математике как ключевым предметам в общеучебной деятельности учащихся.

«Особое значение пропедевтического изучения информатики в начальной школе связано с тем, что в этом курсе есть логически сложные разделы, требующие для успешного освоения развитого логического, алгоритмического, системного мышления. Мы полагаем, что в курсе информатики и ИКТ для начальной школы наиболее целесообразно сконцентрировать основное внимание на развитии мышления школьников и на освоении ими практической работы на компьютере» [2].

Приведенные материалы конкурса «Бобер» показывают, что их можно успешно использовать на уроках информатики в младшей школе для эффективного развития логического, алгоритмического и системного мышления. Так как в базе конкурса уже имеется значительное количество задач, то их можно использовать для организации мини-конкурсов в школе. Возможно, многим ребятам после этого захочется попробовать свои силы для участия в самом конкурсе «Бобер».

ПРИМЕРЫ ЗАДАЧ КОНКУРСА, КОТОРЫЕ МОГУТ БЫТЬ ИСПОЛЬЗОВАНЫ НА УРОКАХ ИНФОРМАТИКИ В ОСНОВНОЙ ШКОЛЕ

Приведем некоторые требования ФГОС к преподаванию информатики в основной школе (ООО) [3].

1. Записывать логические выражения, составленные с помощью операций «и», «или», «не» и скобок, определять истинность такого составного высказывания, если известны значения истинности входящих в него элементарных высказываний.

2. Определять количество элементов во множествах, полученных из двух или трех базовых множеств с помощью операций объединения, пересечения и дополнения.

3. Использовать терминологию, связанную с графами (вершина, ребро, путь, длина ребра и пути), деревьями (корень, лист, высота дерева) и списками (первый элемент, последний элемент, предыдущий элемент, следующий элемент; вставка, удаление и замена элемента).

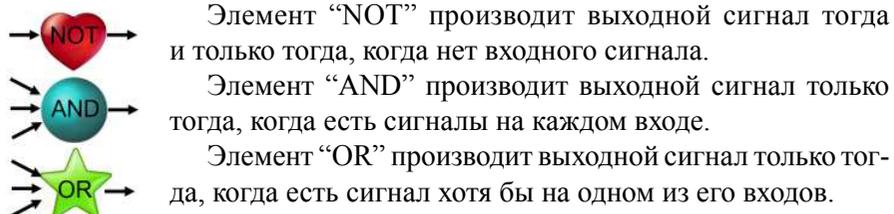
4. Описывать граф с помощью матрицы смежности с указанием длин ребер (знание термина «матрица смежности» не обязательно).

5. Ознакомиться с двоичным кодированием текстов и с наиболее употребительными современными кодами.

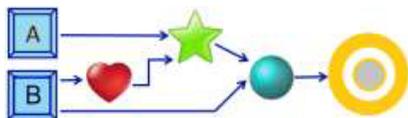
6. Использовать основные способы графического представления числовой информации, (графики, диаграммы).

Задача 1. Электронный замок. Украина

Для защиты своего дома бобёр Биви решил установить электронный замок. Чтобы открыть дверь, Биви должен нажать одновременно определенную комбинацию кнопок. В схеме замка используются три типа элементов, которые преобразуют входные сигналы в один выходной сигнал:

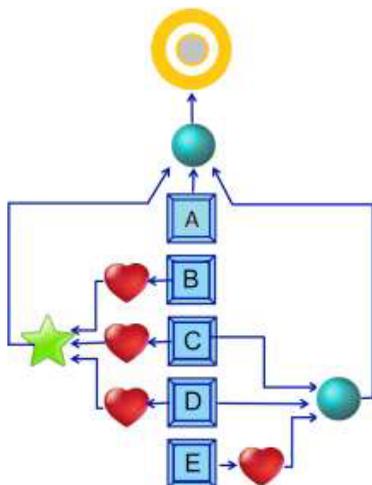


Например, на следующем рисунке замок сработает только, если Вы нажмете кнопку В.



Действительно, запишем эту схему в виде логического выражения: **(А или не В) и В**, следовательно, выражение будет истинно, если истинно В (в скобке автоматически выберется А).

Какая комбинация кнопок откроет замок на этом рисунке?



Прежде всего, нужно предложить записать логическое выражение электронной схемы замка, используя заданные типы элементов. Рассмотрим сигналы справа. Очевидно, что на выход поступают сигналы от кнопок С и D, не поступает сигнал от кнопки Е. Теперь рассмотрим сигналы слева, там не должен поступать сигнал, по крайней мере, из одной из кнопок В, С или D. Следовательно, логическую схему можно записать так: **А и (не В или не С или не D) и (С и D и не Е)**. Это выражение истинно тогда и только тогда, когда А истинно, В ложно (из скобки можно выбрать только не В), С истинно, D истинно, Е ложно, следовательно, для того чтобы замок открылся, нужно нажать кнопки А, С и D.

Логические выражения используются, например, в теории автоматов.

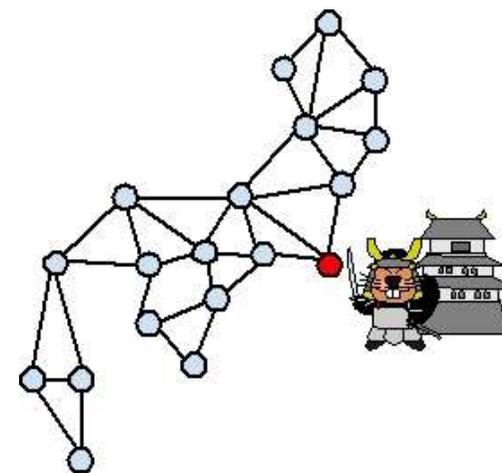
Ответ: А, С и D.

Задача 2. Сигнальные костры. Япония

Давным-давно в Японии на службе правительства сегуната были ниндзя. В случае крайней необходимости они использовали дымовые сигналы, чтобы общаться друг с другом.

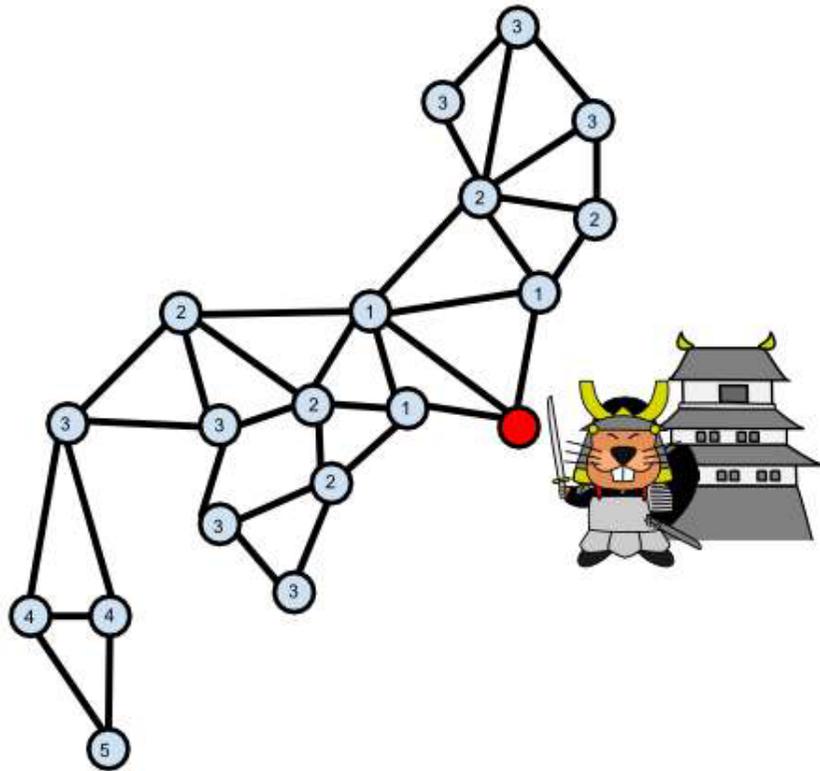
На рисунке красным кружком отмечено правительство сегуната, синим – места, где подаются дымовые сигналы. Кроме того, пункты соединяются линией, если их сигналы могут быть замечены друг от друга (такие пункты называются соседними). В каждом пункте есть ниндзя, который стоит на посту на целый день. Он подаёт дымовой сигнал спустя 1 минуту, после того как увидит сигнал на одном из соседних пунктов (соединенных с ним отрезками на рисунке).

Из правительства сегуната подан сигнал. Через какое время сигналы будут поданы со всех пунктов?



Эта задача доступна не только в основной, но и в младшей школе. Достаточно в каждом кружочке указать, через сколько минут туда придет сигнал.

Сигналы в пунктах, соседних с сегунатом, будут поданы через 1 минуту, поэтому в соответствующих кружочках напишем “1”. Через 2 минуты сигналы будут поданы с пунктов, соседних с предыдущими, в соответствующих кружочках напишем “2” и так далее.



Для решения подобных задач используется теория графов. Граф – способ представления связей между объектами (в данном случае это дымовые сигналы). Объекты представляются вершинами, а связи – рёбрами графа. Графы используются для представления любых видов сетей, например, сетей коммуникаций.

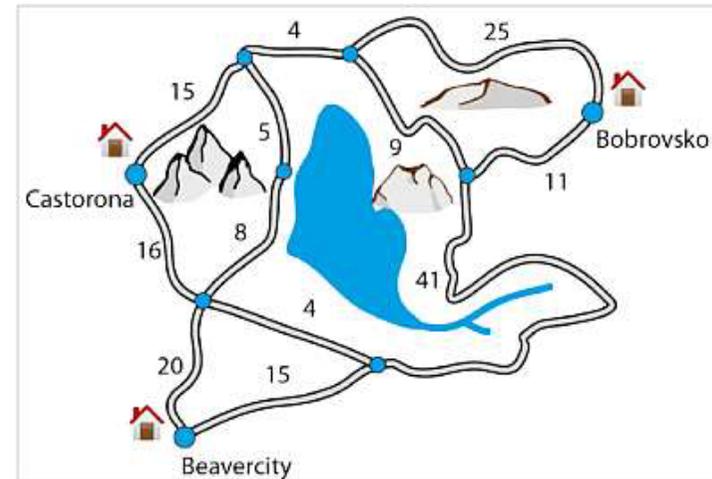
Ответ: 5 минут

Похожая, но более сложная задача, которую можно использовать на уроке информатики в основной школе.

Задача 3. Путешествие бобра. Польша

Бобёр Грег хочет совершить путешествие из Биверсити в Бобровско, а потом в город Касторона. На карте голубыми кружками обозначены маленькие деревни, в которых у Грега есть друзья. Числа рядом с каждой дорогой обозначают время, необходимое Грегу, чтобы пройти по ней. Грег хочет закончить свое путешествие как можно быстрее, но он тратит по 2 минуты на беседу с друзьями в каждой деревне, через которую он проходит.

Сколько времени ему нужно, чтобы дойти до Кастороны, если он задержался в Бобровско на 30 минут (см. рисунок)?



Это задача на нахождение кратчайших путей в графе.

Дорога из Биверсити в Бобровско может быть выбрана так:

1. Рассматриваем путь из Биверсити в соседние точки: это $20+2=22$ или $15+2=17$.

Присваиваем этим точкам значения 22 и 17 соответственно.

2.1. Рассматриваем все дальнейшие пути из точки 22: $22 + 16 + 2 = 40$, $22 + 8 + 2 = 32$.

2.2. Рассматриваем все дальнейшие пути из точки 17: $17 + 41 + 2 = 60$, $17 + 4 + 2 = 23$.

Присваиваем этим точкам метки 40, 32 и 60. Метку 23 мы не присваиваем, так как эта точка уже была с меньшей меткой 22, и путь в нее за 23 минуты мы отбрасываем.

3.1. Продолжаем движение из точки 40 (Касторона): $40 + 15 + 2 = 57$.

3.2. Из точки 32: $32 + 5 + 2 = 39$.

3.3. Из точки 60: $60 + 11 + 30 = 101$ (мы пришли в Бобровско).

Пути 3.1 и 3.2 привели в одну точку. Присваиваем ей меньшее значение 39.

4. Продолжаем путь из точки 39 в соседнюю точку: $39 + 4 + 2 = 45$.

Переходим из точки 45 в соседние точки:

5.1. $45 + 9 + 2 = 56$ (эта точка имеет уже значение 60). Так как $56 < 60$, то изменяем ее значение на 56.

5.2. Из точки 45: $45 + 25 + 30 = 100$ (пришли в Бобровско).

6. Наконец, из точки 56 пришли в Бобровско: $56 + 11 + 30 = 97$.

Таким образом, наименьшее время из Биверсити в Бобровско равно 97.

Теперь нужно пройти путь из Бобровско в Касторону. Но мы его уже проходили двумя путями за $15 + 2 + 4 + 2 + 25 + 2 = 48$ минут и $15 + 2 + 4 + 2 + 9 + 2 + 11 = 45$ минут.

Окончательный вариант $97 + 45 + 2 = 144$ минуты.

Ответ: 144 минуты.

Замечание. Описанное решение – это применение алгоритма Дейкстры, который используется в задачах о нахождении кратчайшего пути. Можно в этом месте рассказать об алгоритме Дейкстры более подробно.

В нашем простом случае можно было просто просчитать все возможные пути и выбрать тот, на который потребуется наименьшее время:

1) $15 + 41 + 11 + 2 + 2 + 30 = 101$

2) $15 + 4 + 8 + 5 + 4 + 9 + 11 + 6 \cdot 2 + 30 = 98$

3) $20 + 8 + 5 + 4 + 9 + 11 + 5 \cdot 2 + 30 = 97$

4) $15 + 4 + 8 + 5 + 4 + 25 + 5 \cdot 2 + 30 = 101$

5) $20 + 8 + 5 + 4 + 25 + 5 \cdot 2 + 30 = 102$

Самый короткий путь из Биверсити в Бобровско можно пройти за 97 минут.

Возможные пути из Бобровско в город Касторона:

1) $25 + 4 + 15 + 2 + 2 = 48$

2) $11 + 9 + 4 + 15 + 3 \cdot 3 = 47$

Выбираем второй путь. Общее время $97 + 47 = 144$.

Ответ: 144 минуты

Задача 4. Склад. Япония

Плотник в Бобровой Деревне использует 31 склад, пронумерованный от 1 до 31. Однажды, он забыл, сколько складов уже заполнил, но помнит, что заполнял их в порядке возрастания номеров.



Чтобы уменьшить количество открывания дверей, он действует следующим образом:

Сначала, открывает склад со средним номером – склад № 16.

Затем:

– если склад № 16 пуст, он решает искать первый незаполненный склад в промежутке от № 1 до № 15, открывает опять средний склад – склад № 8 – и повторяет процедуру;

– если склад № 16 заполнен, то нужный склад он ищет между № 17 и № 31, открывает средний склад – склад № 24 – и повторяет процедуру.

После всех действий плотник обнаружил, что заполнены были склады от № 1 до № 15 включительно. Сколько дверей ему пришлось открыть?

Здесь мы используем так называемый двоичный поиск, который применяется к упорядоченному набору, каждый элемент которого обладает или не обладает некоторым свойством. Это важный базовый алгоритм.

Выберем склад, находящийся посередине. Если он не заполнен, то искать нужно слева от него, если заполнен, то справа. Посередине находится склад с номером 16. Так как оказалось, что последним заполненным был склад с номером 15, то склад с номером 16 был пуст. Плотник выбрал из 15 складов справа склад с номером 8 (средний склад), он оказался заполнен. Теперь нужно было искать последний заполненный склад среди складов с номерами 9–15. Таких складов 7. Выбираем средний с номером 12, он также оказался заполнен. Теперь слева от него склады с номерами 13–15, выбираем склад посередине с номером 14. Осталось проверить склад с номером 15. Итого 5 открываний дверей с номерами 16, 8, 12, 14 и 15.

Ответ: 5.

**ПРИМЕРЫ ЗАДАЧ КОНКУРСА,
КОТОРЫЕ МОГУТ БЫТЬ ИСПОЛЬЗОВАНЫ
НА УРОКАХ ИНФОРМАТИКИ
В СТАРШЕЙ ШКОЛЕ**

В числе требований ФГОС к преподаванию информатики в старшей школе (ООП ССО) [4]:

1. Овладение понятием сложности алгоритма, знание основных алгоритмов обработки числовой и текстовой информации, алгоритмов поиска и сортировки.

2. Владение универсальным языком программирования высокого уровня (по выбору), представлениями о базовых типах данных и структурах данных; умение использовать основные управляющие конструкции.

3. Владение навыками и опытом разработки программ в выбранной среде программирования, включая тестирование и отладку программ; владение элементарными навыками формализации прикладной задачи и документирования программ.

4. Сформированность представлений о важнейших видах дискретных объектов и об их простейших свойствах, алгоритмах анализа этих объектов, о кодировании и декодировании данных и причинах искажения данных при передаче; систематизацию знаний, относящихся к математическим объектам информатики; умение строить математические объекты информатики, в том числе логические формулы.

Использование заданий конкурса «Бобер» дает возможность сделать изучение соответствующих тем интересным и эффективным. Проверить свои знания и возможности можно, приняв участие в конкурсе «Бобер» в условиях ограниченного времени принятия решений.

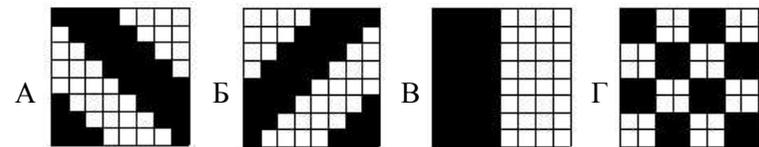
Задача 1. Боберитовый формат.  Россия

Бобёр Хакро придумал способ кодировки черно-белых изображений. Каждое изображение он кодирует перечислением цветов: сначала слева направо, затем сверху вниз. Например, изображение на рисунке

 он кодирует как WWWBWW.

Затем Хакро решил улучшить метод, и, если буква повторяется несколько раз подряд, он пишет ее только один раз, сопровождая числом повторений. Он назвал этот формат BVR. В этом формате приведенное изображение кодируется всего 6 символами: W4B3W2.

Какая из следующих картинок будет иметь самый короткий код в формате BVR?



Непосредственный подсчет показывает, что самый короткий код в формате BVR имеет картинка А: B4W5B4W5B4W5B4W5B5W4B5W4B5W4B (14 блоков из двух символов и один блок из одного символа). Картинка Б имеет код: W4B4W3B4W3B4W3B4W3B4W4B3 W4B3W4B3 (18 блоков из двух символов). Картинка В имеет код: B4W4B4W4B4W4B4W4B4W4B4W4B4W4 (16 блоков из двух символов). Наконец, картинка Г имеет код: B2W2B2W2B2W2B2W2W2B2W2B2W2B2W2B2W2B2W2B2W2W2B2W2B2W2 (32 блока из двух символов).

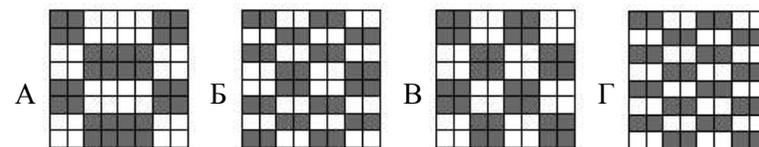
Ответ: рисунок А.

Задача 1'. Формат Боберитовый-2.  Россия

Но затем Хакро решил улучшать формат дальше, теперь он записывает не только повторение букв, но и блоков из нескольких букв, заключая их в скобки. Этот формат он назвал BVR2. Например, картинка  кодируется в новом формате так:

4 (BW)B.

Какая из следующих картинок записывается в формате BVR2 как 2(4(2B2W)4(2W2B))?



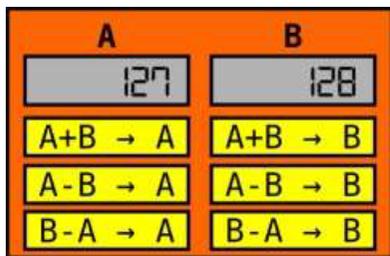
Очевидно, что должно быть два одинаковых блока. Этому требованию не удовлетворяет рисунок Б. Далее нужно выбрать рисунок, в котором повторяются блоки 2B2W. Этому требованию удовлетворяют рисунки В и Г, но на рисунке Г таких идущих подряд блоков – 2, а нужно, чтобы их было 4. Остается блок В. Проверяем следующие блоки в верхней половине: 4 подряд идущих блока 2W2B, то есть 4(2W2B). Таким образом, двум верхним строчкам, а, следовательно, и всему условию, соответствует рисунок В.

Ответ: В.

В предыдущей задаче в формате DVR2 рисунок А можно было закодировать так: 4(B4W5)3(B5W4)B, код рисунка Б имел бы вид: W4B44(W3B4)4(W4B3) и т. д.

Задача демонстрирует основную идею сжатия изображения, которое используется в некоторых современных форматах графических файлов. Самые известные из них GIF и PNG.

Задача 2. Бобровый калькулятор. Россия



Калькулятор бобра имеет два дисплея (А и В) и шесть кнопок. Кнопки позволяют складывать и вычитать числа на дисплеях и сохранять результат на одном из дисплеев.

Какая последовательность действий приводит к обмену чисел между дисплеями?

Эта задача имеет не единственное решение.

Например:

1 шаг: $A + B \rightarrow A$ (в результате на дисплее А появилась сумма чисел, а на дисплее В число осталось без изменения).

2 шаг: $A - B \rightarrow B$ (теперь на дисплее А осталась сумма чисел, а на дисплее В – то число, которое первоначально было на А).

3 шаг: $A - B \rightarrow A$. Исходные значения дисплеев обменялись.

Пример другого решения:

Пусть $A > B$.

1 шаг $A - B \rightarrow B$ (на дисплее А число не изменилось, на дисплее В появилась разность чисел).

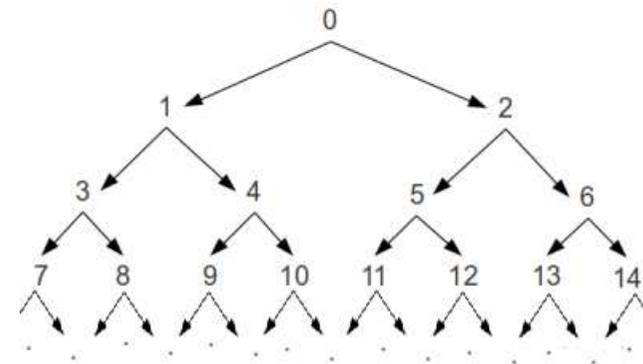
2 шаг $A - B \rightarrow A$ (на дисплее А появится число, которое первоначально было на В, а на дисплее В останется разность первоначальных чисел $A - B$).

3 шаг: $A + B \rightarrow B$. Исходные значения дисплеев обменялись.

Дисплеи в этой задаче играют роль переменных, которые мы используем, чтобы хранить в них некоторую текущую информацию в программах. Если нужно воспользоваться информацией, мы называем имя переменной, в которой она хранится.

Представленная задача являлась классической для программистов, когда нужно было как можно больше экономить память вычислительной машины.

Задача 3. Спуск по дереву. Канада



Целые числа, начиная с 0, расположены, как показано на рисунке. Под каждым числом находятся два числа: одно – слева, другое – справа. Например, под числом 3 число 7 находится слева, а 8 справа. Числа располагаются в порядке возрастания сверху вниз, а внутри каждого ряда – слева направо.

Чтобы дойти от 0 до числа 11, нужно сначала пойти направо, а потом два раза налево.

Какая последовательность переходов налево (обозначаются L) и направо (обозначаются R) приведет от 0 к 100?

Заметим, что:

1. Ряд с номером k начинается с числа $2^{k-1} - 1$ и заканчивается удвоенным числом $2(2^{k-1} - 1)$:

$k = 1 \rightarrow 2^{1-1} - 1 = 0, k = 2 \rightarrow 2^{2-1} - 1 = 1, k = 3 \rightarrow 2^{3-1} - 1 = 1$ и т. д.

Число 100 находится в 7 ряду, так как $2^{7-1} - 1 < 100 < 2(2^{7-1} - 1)$.

2. Четные числа всегда достигаются при спуске вправо, следовательно, последней буквой должно быть R.

3. К нечетному числу $2n + 1$ мы спускаемся от расположенного выше числа n влево.

Теперь находим последовательность чисел в обратном порядке:

Число 100 находится в паре с числом 99, следовательно, в шестом ряду над ними находится число $(99 - 1) : 2 = 49$, при этом от числа 49 мы спускаемся к числу 100 вправо (R).

К числу 49 спускаемся из пятого ряда от числа $(49 - 1) : 2 = 24$ влево (L).

Число 24 находится в паре с числом 23, а над ними в четвертом ряду находится число

$(23 - 1) : 2 = 11$, от которого мы спускаемся к числу 24 вправо (R).

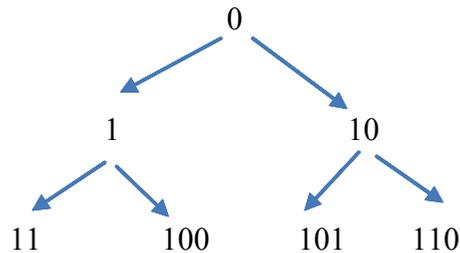
К числу 11 «дорога» известна: RLL.

Теперь объединяем спуски от 0 до 11 и от 11 до 100: RLLRLR.

Ответ: RLLRLR

Задачу можно было бы решать, записав все числа в двоичной системе.

Замечание. Запишем все числа в двоичной системе:



и т. д.

Четные числа заканчиваются нулем, нечетные –1. Число над парой получаем следующим образом: в нечетном числе отбрасываем последнюю 1.

Число 100 в двоичной системе имеет вид: **1100100**. Слева стоит число 1100011, число сверху – **110001** (нечетное). Спуск от него к числу 1100010 вправо (R). Над числом 110001 стоит число **11000** (четное), спуск от него к числу 110001 влево (L). Слева от числа 11000 стоит нечетное число 10111. Над ними сверху находится число **1011** (нечет-

ное), от которого спускаемся к числу 11000 вправо (R). К числу 1011 спускаемся от числа **101** (нечетное) влево (L). К числу 101 спускаемся от числа **10** влево (L). К числу 10 спускаемся **0** вправо (R).

Таким образом, спуск выглядит так (идем от конца к началу):

$0 \rightarrow 10 \rightarrow 101 \rightarrow 1011 \rightarrow 11000 \rightarrow 110001 \rightarrow 1100100$,

то есть получили последовательность букв RLLRLR.

Заметим, что каждая строка с номером k начинается с числа, состоящего из $k - 1$ единицы, а заканчивается числом из того же количества единиц и нуля, то есть число $2^k - 1$ записывается k единицами.

В задаче представлена определенная структура связи «предков» и «потомков», которая представляет собой бинарное дерево. Такая структура в информатике называется «куча».

Эта несложная задача, тем не менее, может быть очень полезна и при изучении двоичной системы счисления, и при изучении вопроса о хранении данных, о построении бинарных деревьев.

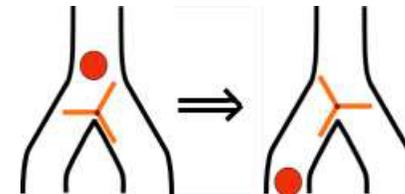
Можно предложить обучающимся такие вопросы:

- В каком ряду находится число m ?
- За сколько шагов можно добраться к числу n ?
- Является ли решение такой задачи единственным? Почему?

Можно в дополнение к этой задаче предложить задачу, близкую к ней, но другую:

Задача 5. Вертушка и шарики. Россия

Шарик, падающий сверху, продолжает свое движение налево или направо в зависимости от состояния «вертушки». При этом каждый шарик изменяет её состояние на противоположное:



Биграм показал надпись коллегам. Никто не мог ее расшифровать, но все согласились, что она написана на древнем марсианском языке.

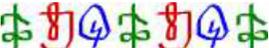
Спустя некоторое время, Биграм нашел еще 4 скалы с надписями на этом языке. Чтобы узнать, подлинны ли эти надписи или они просто стилизованы под древнюю марсианскую надпись, Биграм использует следующий метод. Для каждой пары символов каждой надписи он выясняет, сколько раз она встречается в самой первой, заведомо настоящей, надписи. Полученные числа он перемножает.

Например, в надписи  он рассмотрел пары  и .

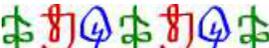
Из этих пар первая встречается в исходной надписи 3 раза, вторая – 2 раза, третья – 1 раз. Произведение $3 \cdot 2 \cdot 1 = 6$.

Подлинной Биграм считает ту надпись, для которой вычисленное произведение наибольшее.

Какую из следующих надписей Биграм считает подлинной?

А  Б 
В  Г 

На самом деле не нужно подсчитывать очки, так как все надписи состоят из почти одинаковых пар.

Закодируем надписи латинскими буквами. Например, первая надпись  превратится в ABCAVCA, вторая надпись – в ABCAVCB, третья – в ABCAVCC, четвертая – в SAVCAVC. Исходная надпись будет иметь вид: ABCACVAVCAAVCBVAV.

Сравниваем надпись А с исходной: АВ встречается в исходной записи 3 раза, ВС – 2 раза, СА – 1 раз, итого $3 \cdot 2 \cdot 1 = 6$.

Сравниваем надпись В с исходной, получаем $3 \cdot 2 \cdot 1 \cdot 3 = 18$ (добавляется пара СВ, которая встречается 3 раза). Сравниваем надпись С с исходной: $3 \cdot 2 \cdot 1 \cdot 0 = 0$, так как пара СС в исходной надписи не встречается. Сравниваем надпись Г с исходной: $1 \cdot 3 \cdot 2 = 6$. Таким образом, подлинной надписью является запись Б.

Ответ: Б.

Задача основана на использовании модели двух грамматик (bi-gram language model). Метод заключается в подсчете числа очков в предложениях, чтобы компьютер мог определить, какие предложения более похожи на предложения естественного языка. Такие модели широко используются почти во всех задачах, связанных с анализом языков, от распознавания речи до машинного перевода.

Мы привели примеры только из конкурса 2013 года, причем только небольшую часть задач.

На сайте можно найти задачи полегче и потруднее. Их можно решать на занятии и задавать домой. Можно решать группами и индивидуально, можно устраивать соревнования между группами и между отдельными учениками. По многим темам, изучаемым на уроках информатики, можно подобрать задачи, которые являются одновременно забавными и в то же время требуют знаний, творческого подхода, сообразительности и быстроты реакции.

Литература

1. В. Дагене. Международный конкурс по информатике и компьютерной грамотности «Bebras» // Компьютерные инструменты в школе, 2013. № 1. С. 43–47.
2. Информатика – Школа 2100. [Электронный ресурс] <http://school2100.com>
3. Примерная основная образовательная программа основного общего образования» одобрена решением федерального учебно-методического объединения по общему образованию (протокол от 8 апреля 2015 г. № 1/15).
4. Примерная основная образовательная программа среднего общего образования, одобрена решением федерального учебно-методического объединения по общему образованию (протокол от 28 июня 2016 г. № 2/16-з). <https://aujc.ru/dokumenty-fgos-uchitelyu-informatic/>