Графы и теория игр: задачи С3, В9 и В13

Д. А. Михалин, учитель информатики ГБОУ ЦО №345

Введение

Несколько заданий из демоверсии ЕГЭ 2013 года связано с графами или построением дерева игры. Это задачи А2, В9, В13 и С3. При этом в задаче В9 граф изображен явно, в задаче С3 требуется построить дерево игры. В двух других задачах про графы явно не сказано, но их использование может упростить решение, а также показывает схожесть различных по формулировкам задач.

История задачи С3

Несколько лет мы решали задачи про двух игроков, которые перекладывали камушки, двигали фишки на координатной плоскости и т.д. К этим задачам было много справедливых претензий. В прошлом году мы увидели совсем новое задание — на подсчет числа возможных программ. Но и эта задача была трудна для проверки (например, была размыта граница 2 и 3 баллов), было не очень понятно, что она проверяла (вряд ли заявленное “умение построить алгоритм”), и от нее тоже отказались.

В этом году мы снова увидели перекладывающих камни Петю и Ваню в задаче, одно только условие которой занимает почти страницу. Сначала это вызвало ропот многих учителей и испуг ребят, но при внимательном рассмотрении выяснилось, что авторы сделали попытку не усложнить задачу, а приблизить задачу на игры к реальным приемам определения выигрышной стратегии: с конца, или начиная с тривиальных случаев. При этом также уменьшается объем вычислительной работы, что снижает вероятность ошибки.

Решение задачи С3

Вот пример задания из демоверсии (условие немного сокращено):

> *Два игрока, Петя и Ваня, играют в следующую игру. Перед игроками лежит куча камней. Игроки ходят по очереди, первый ход делает Петя. За один ход игрок может добавить в кучу один камень или увеличить количество камней в куче в два раза.*
>
> *Игра завершается в тот момент, когда количество камней в куче становится не менее 22. Победителем считается игрок, сделавший последний ход, то есть первым получивший кучу, в которой будет 22 или больше камней. В начальный момент в куче было S камней, 1 ≤ S ≤ 21.*
>
> *Выполните следующие задания. Во всех случаях обосновывайте свой ответ.*
>
> *1. а) Укажите все такие значения числа S, при которых Петя может выиграть в один ход. Обоснуйте, что найдены все нужные значения S, и укажите выигрывающий ход для каждого указанного значения S.*
> *б) Укажите такое значение S, при котором Петя не может выиграть за один ход, но при любом ходе Пети Ваня может выиграть своим первым ходом. Опишите выигрышную стратегию Вани.*
>
> *2. Укажите два таких значения S, при которых у Пети есть выигрышная стратегия, причём*
> *– Петя не может выиграть за один ход, и*
> *– Петя может выиграть своим вторым ходом, независимо от того, как будет ходить Ваня.*
> *Для каждого указанного значения S опишите выигрышную стратегию Пети.*
>
> *3. Укажите значение S, при котором:*
> *– у Вани есть выигрышная стратегия, позволяющая ему выиграть первым или вторым ходом при любой игре Пети, и*
> *– у Вани нет стратегии, которая позволит ему гарантированно выиграть первым ходом.*
> *Для указанного значения S опишите выигрышную стратегию Вани. Постройте дерево всех партий,*

возможных при этой выигрышной стратегии Вани (в виде рисунка или таблицы). На рёбрах дерева указывайте, кто делает ход, в узлах – количество камней в куче.

В демоверсии присутствует авторское решение этой задачи. Мы приведем его в несколько другой форме: запишем в первой строке таблицы значения S от 1 до 21, а во второй строке будем указывать, является ли эта позиция выигрышной (В) или проигрышной (П) для того игрока, который из этой позиции делает ход.

Тут следует сразу отметить принципиальный момент, который все ребята должны усвоить:

- *если в некоторой исходной позиции есть **хотя бы один** ход, который приводит игру в проигрышную позицию, то исходная позиция является выигрышной;*
- *если в некоторой исходной позиции **любой** ход приводит игру в выигрышную позицию, то исходная позиция является проигрышной.*

Заметим, что если нам нужно найти значения S, для которых выигрывает первый игрок, то мы ищем выигрышные позиции — в них у игрока есть выигрышная стратегия при любых ходах противника. И наоборот, если мы ищем значения S, для которых выигрывает второй игрок, то мы ищем проигрышные позиции — в них первый игрок проиграет (при правильной игре противника), как бы он сам не ходил.

Теперь начнем заполнять таблицу. Решение п.1а очевидно:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
										В	В	В	В	В	В	В	В	В	В	В

Для ответа на вопрос 1б необходимо найти проигрышную позицию, т.е. такую, из которой любой ход ведет в выигрышную позицию. Мы будем обозначать ходы стрелками (для удобства, прибавление камня — стрелкой сверху, удвоение — стрелкой снизу). Эта позиция – S=10.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
									П	В	В	В	В	В	В	В	В	В	В	В

Теперь, когда мы нашли проигрышную позицию, мы можем ответить на вопрос 2, в котором требуется найти еще две выигрышные позиции, т.е. такие, из которых хотя бы один ход ведет в проигрышную позицию. Это S=5 и S=9.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
				В				В	П	В	В	В	В	В	В	В	В	В	В	В

В п.3 нам нужно найти еще одну проигрышную позицию, из которой любой ход ведет в найденные ранее выигрышные позиции. Конечно, она не обязательно будет единственной, как S=8 в нашем случае:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
				В			П	В	П	В	В	В	В	В	В	В	В	В	В	В

Дерево игры повторять не будем, оно построено в демоверсии двумя способами: графическим и табличным. Отметим лишь, что все его ребра, кроме последних (ведущих за границу нашей таблицы), мы уже указали на разных этапах решения.

Возможные ошибки

К сожалению, сейчас сложно говорить о типичных ошибках, так как реальные решения задания в нынешней формулировке участниками ЕГЭ пока почти никто из экспертов не видел. Мы можем ориентироваться или на предыдущий опыт, или на то, что видим у наших собственных учеников.

1. Указание не всех возможных ходов в проигрышных позициях (отсюда, возможно, учет проигрышной позиции как выигрышной). Такая ошибка встречалась в работах позапрошлого года (и предыдущих). Следует добиться полного понимания учащимися выделенной выше связи между выигрышными и проигрышными позициями.
2. Недостаточные обоснования. Раньше в указаниях к оцениванию встречались фразы вроде «строгое доказательство», «полное обоснование», что не всегда поддавалось оценке на реальных работах. Сейчас указания написаны более конкретно, но это не освобождает учащихся от необходимости объяснять, почему та или иная позиция является выигрышной или проигрышной. Пример обоснования указан в демоверсии.
3. Арифметические ошибки, приводящие к *неправильному* ответу. Хотя за ошибки, приводящие к правильному ответу, баллы не снимаются, раньше довольно часто встречались ошибки, приводящие к неправильному ответу. И было иногда очень обидно ставить 0 баллов за решение, из которого видно полное понимание автором того, что нужно делать, но в котором присутствовала арифметическая ошибка в самом начале.

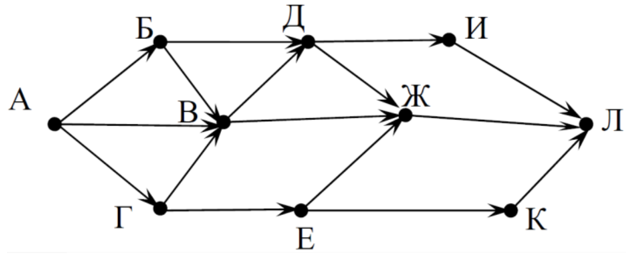
Другие задачи на графы

Упомянутую задачу А2 мы рассматривать не будем, она достаточно простая (но стоит, конечно, отметить, что при переходе от таблицы весов к графическому изображению графа число ошибок снижается).

Задача В9

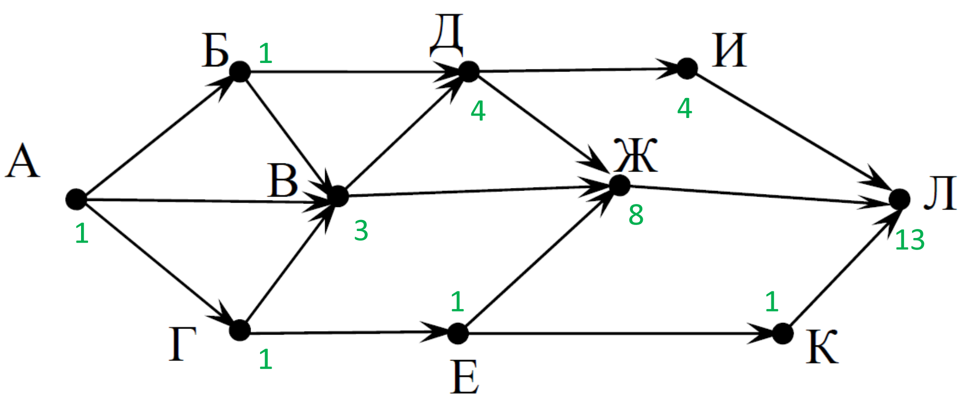
В9 — задача на подсчет числа путей в графе:

> *На рисунке – схема дорог, связывающих города А, Б, В, Г, Д, Е, Ж, И, К, Л. По каждой дороге можно двигаться только в одном направлении, указанном стрелкой. Сколько существует различных путей из города А в город Л?*



Эту задачу можно решать разными способами: полным алфавитным перебором путей, табличным способом. Рассмотрим графический способ решения.

Будем отмечать рядом с вершинами графа число путей, которыми можно в них попасть из А. При этом важно понимать, что в саму вершину А можно попасть одним способом: никуда не ходить (пустая программа). Далее, если в некоторую вершину Щ ведут пути из вершин $Ц_1$, …, $Ц_k$, то число путей из А в Щ, которое мы обозначим N(Щ), равно $N(Ц_1)+…+N(Ц_k)$. Поэтому мы должны подписывать значения около вершин в таком порядке: сначала А, потом вершины, в которые ведет только одна дорога — из А (в нашем примере это Б и Г), далее — те вершины, в которые ведут дороги только из уже подписанных вершин. В итоге, мы получаем граф, из которого видно, что в вершину Л ведет 13 путей из А.



Задача B13

У исполнителя Удвоитель две команды, которым присвоены номера:

1. *прибавь 1,*
2. *умножь на 2.*

Первая из них увеличивает на 1 число на экране, вторая удваивает его. Программа для Удвоителя – это последовательность команд. Сколько есть программ, которые число 3 преобразуют в число 23?

B13 — это, фактически, задача C3 из вариантов 2012 года, в которой больше не требуется запись решения (что в разы сокращает время на ее выполнение). Но самое интересное, что это, на самом деле, та же задача, что и B9: в обоих случаях необходимо подсчитать число путей в ориентированном графе. Опять же, эту задачу можно решать по-разному. Приведем один из возможных вариантов решения.

Для удобства изображения графа, выпишем его вершины (т.е. возможные числа на экране) в ряд, как в задаче C3. Ниже вершин, или во второй строке таблицы, мы будем указывать число путей, ведущих в эту вершину из исходной вершины (или число программ). Стрелками, разумеется, обозначим возможные переходы между вершинами, соответствующие командам исполнителя.

Число на экране	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Число путей																					

Пути, соответствующие команде «умножь на 2», но ведущие за границу нашей таблицы, не показаны: они нас не интересуют, т.к. в рассматриваемой задаче число на экране не может уменьшаться.

На практике совершенно не обязательно изображать стрелки, соединяющие соседние числа и соответствующие команде «прибавь 1». Здесь они даны для лучшего понимания того факта, что изображен граф, а задача совпадает с предыдущей. Можно вместо стрелок внести в таблицу еще одну строку, в которой для каждого числа указать, из каких чисел его можно получить.

Снова мы подписываем число путей, начиная с первой вершины (такой путь один — это пустая программа). Дальше нам даже не нужно заботиться о правильном порядке подписывания вершин — т.к. число не может уменьшаться, мы просто подписываем число путей у всех вершин по очереди все по тому же правилу, как в задаче B9. Теперь формула выглядит так: $N(k)=N(k-1)+N(k/2)$, где $k>3$, а $N(k/2)$ принимается равным нулю, если k не делится на 2. Получаем следующую таблицу:

Число на экране	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Число путей	1	1	1	2	2	3	3	4	4	6	6	8	8	11	11	14	14	18	18	22	22

Таким образом, существует 22 программы, которые преобразуют число 3 в число 23.

Мы убедились, что задачи B9 и B13 решаются одинаково, хоть и выглядеть это решение может по-разному.

Список литературы

1. Демонстрационный вариант КИМ ЕГЭ 2013 года по информатике и ИКТ. Спецификация КИМ для проведения в 2013 году ЕГЭ по информатике и ИКТ (http://fipi.ru/binaries/1384/inf11.zip)
2. Учебно-методические материалы для председателей и членов региональных предметных комиссий по проверке выполнения заданий с развернутым ответом экзаменационных работ ЕГЭ 2013 года. Информатика и ИКТ (http://fipi.ru/binaries/1466/UMMinf.zip)
3. К. Ю. Поляков. ЕГЭ: новые стратегии (задача C3) // Информатика, № 1, 2013, с. 22-27.