Дискретная математика. Домашнее задание 22 (повторение), решения

1. Найдите максимальное количество рёбер в таких ориентированных графах на n вершинах, которые не имеют ориентированных циклов.

Решение. Между двумя вершинами u и v возможны два ориентированных ребра: (u, v) и (v, u). Вместе они образуют ориентированный цикл длины 2. Поэтому в орграф без ориентированных циклов может входить не более одного ребра для каждой пары вершин. Поэтому количество рёбер не больше $\binom{n}{2}$.

Построим пример орграфа без ориентированных циклов с $\binom{n}{2}$ рёбрами. Пусть вершины занумерованы числами от 1 до n. Рёбра имеют вид (i, j), $i < j$. Количество рёбер в таком графе $\binom{n}{2}$. Ориентированных циклов нет, так как вдоль любого ориентированного пути номера вершин возрастают.

Ответ: $\binom{n}{2}$.

2. Случайно, равновероятно и независимо выбираются два положительных делителя числа 12!. Сравните вероятность того, что эти делители взаимно просты, с 1/100.

Решение. Разложение 12! на простые множители имеет вид:

$$12! = 2^{10} \cdot 3^5 \cdot 5^2 \cdot 7 \cdot 11.$$

Делители 12! имеет вид $2^{a_2}3^{a_3}5^{a_5}7^{a_7}11^{a_{11}}$, где a_i принимают следующие значения:

$$0 \le a_2 \le 10; \quad 0 \le a_3 \le 5; \quad 0 \le a_5 \le 2; \quad 0 \le a_7 \le 1; \quad 0 \le a_{11} \le 1.$$

Равновероятный выбор делителя 12! означает, что мы случайно, равновероятно и независимо выбираем каждое из a_i из указанных интервалов. Два делителя с показателями a_i, b_i взаимно просты, если для каждого i выполняется $\min(a_i, b_i) = 0$.

Найдём вероятность p события $\min(a, b) = 0$, если a, b выбираются случайно и независимо из интервала от 0 до k. Всего исходов $(k+1)^2$. Благоприятные исходы: $a = 0$, b любое (таких $k+1$ штука) или a любое, $b = 0$ (таких тоже $k+1$ штука). В пересечении этих множеств есть ровно один исход $a = 0$, $b = 0$. Поэтому

$$p = \frac{2(k+1) - 1}{(k+1)^2} = \frac{2k+1}{(k+1)^2}\ .$$

Для $i = 2, 3, 5, 7, 11$ обозначим вероятность события $\min(a_i, b_i) = 0$ через p_i. Получаем

$$p_2 = \frac{21}{121}; \quad p_3 = \frac{11}{36}; \quad p_5 = \frac{5}{9}; \quad p_7 = p_{11} = \frac{3}{4}.$$

Поскольку события, которые задаются условиями $\min(a_i, b_i) = 0$, независимы, то вероятность, что два случайных делителя 12! взаимно просты, равна произведению этих чисел

$$\frac{21}{121} \cdot \frac{11}{36} \cdot \frac{5}{9} \cdot \frac{3}{4} \cdot \frac{3}{4} > \frac{20}{120} \cdot \frac{10}{35} \cdot \frac{4}{8} \cdot \frac{3}{4} \cdot \frac{3}{4} = \frac{90}{6 \cdot 35 \cdot 2 \cdot 16} = \frac{15}{70 \cdot 16} = \frac{3}{14 \cdot 16} = \frac{3}{224} > \frac{1}{100}\ .$$

Ответ: искомая вероятность больше 1/100.

3. Найдите НОД($\underbrace{11\ldots11}_{120 \text{ штук}}, \underbrace{11\ldots11}_{84 \text{ штуки}}$). (Числа записаны в десятичной системе счисления.)

Решение. Суммируя геометрическую прогрессию, получаем

$$\underbrace{11\ldots11}_{k \text{ штук}} = 1 + 10 + \ldots + 10^{k-1} = \frac{10^k - 1}{9}\ .$$

Применяя несколько раз равенство НОД$(a,b) =$ НОД$(a - qb, b)$ при подходящих $q = 10^s$, получаем

$$\text{НОД}(\tfrac{1}{9}(10^{120}-1), \tfrac{1}{9}(10^{84}-1)) = \text{НОД}(\tfrac{1}{9}(10^{36}-1), \tfrac{1}{9}(10^{84}-1)) = \text{НОД}(\tfrac{1}{9}(10^{36}-1), \tfrac{1}{9}(10^{48}-1)) =$$
$$= \text{НОД}(\tfrac{1}{9}(10^{36}-1), \tfrac{1}{9}(10^{12}-1)) = \text{НОД}(\tfrac{1}{9}(10^{36}-1), \tfrac{1}{9}(10^{12}-1)) =$$
$$= \text{НОД}(\tfrac{1}{9}(10^{24}-1), \tfrac{1}{9}(10^{12}-1)) = \text{НОД}((10^{12}+1)\cdot\tfrac{1}{9}(10^{12}-1), \tfrac{1}{9}(10^{12}-1)) = \frac{10^{12}-1}{9}\ .$$

Ответ: $\underbrace{11\ldots11}_{12 \text{ штук}}$.

Замечание: равенство $12 =$ НОД$(120, 84)$ не случайно. В проделанных выше вычислениях легко увидеть вычисление наибольшего общего делителя показателей.

4. Рассмотрим вход функции Tree: $\{0,1\}^{\binom{n}{2}} \to \{0,1\}$ как неориентированный граф на n вершинах и положим Tree$(G) = 1$ тогда и только тогда, когда G дерево. Найдите сложность вычисления функции Tree в модели разрешающих деревьев.

Решение. Как было показано на лекциях, сложность булевой функции в модели разрешающих деревьев не превышает числа ее переменных. Таким образом, в нашем случае сложность не больше $\binom{n}{2}$.

Докажем, что сложность не меньше этого значения. Предположим, что есть разрешающее дерево глубины меньше $\binom{n}{2}$. Опишем стратегию противника, при которой разрешающее дерево даёт для некоторого графа неверный ответ.

Обозначим через G_k граф, рёбра которого отвечают тем переменным, на запрос значений которых был дан ответ 1 после k запросов. В корне разрешающего дерева рёбер нет вообще, поэтому G_0 состоит из изолированных вершин (степень 0). Этот граф — лес (каждая компонента связности — дерево).

Стратегия противника состоит в том, чтобы поддерживать этот граф в виде леса и по возможности уменьшать количество компонент связности. Другими словами, если запрос относится к ребру, соединяющему две разные компоненты текущего графа G_k, то ответ 1, а если к ребру внутри одной компоненты связности, то ответ 0.

Рассмотрим лес, который отвечает листу разрешающего дерева. Пусть он несвязный. Ни одно из рёбер между компонентами связности не было спрошено (так как в противном случае оно вошло бы в лес по построению стратегии). Таким образом, в этот лист попадают как деревья (добавим нужное количество рёбер между компонентами связности), так и не деревья (скажем, сам граф $G_{\text{лист}}$). Поскольку значение функции в этих случаях разное, в одном из случаев разрешающее дерево выдаст неверный ответ.

Пусть в листе получилось остовное дерево. Какое-то из рёбер не было спрошено (так как вопросов меньше $\binom{n}{2}$)). Если значение соответствующей переменной равно 1, то граф не дерево, а если 0, то может быть деревом (все значения переменных равны 0, кроме тех, которым уже присвоена 1). Опять-таки, в одном из случаев ответ будет неверный.

Ответ: $\binom{n}{2}$.

5. Булева функция $U_2(x_1, \dots, x_n)$ равна 1 тогда и только тогда, когда среди значений $x_1, \dots, x_n$ есть ровно 2 единицы. Постройте схему размера $O(n)$, вычисляющую функцию U_2.

Решение. Введём помимо входных переменных x_i ещё три группы переменных: y_i («среди первых i переменных x_j есть хотя бы одна 1»), z_i («среди первых i переменных x_j есть хотя бы две 1»), u_i («среди первых i переменных x_j есть хотя бы три 1»). Здесь i пробегает значения от 1 до n.

Для простоты записи схемы введём также две вспомогательные переменные $y_0 = z_0 = u_0 = 0$. Эти переменные вычисляются следующей схемой (последовательностью присваиваний, начальный отрезок из переменных пропущен):

$$g_0 := \neg x_1;\ y_0 := x_1 \wedge g_0;\ z_0 := x_1 \wedge g_0;\ u_0 := x_1 \wedge g_0;$$

поскольку $x \wedge \neg x = 0$ при любом значении булевой переменной x. Эта схема является началом схемы вычисления функции U_2. Дальнейшие присваивания разбиваются на группы, в i-й группе присваиваний вычисляются y_i, z_i и u_i:

$$y_i := x_i \vee y_{i-1};\ g_{2i-1} := x_i \wedge y_{i-1};\ z_i := z_{i-1} \vee g_{2i-1};\ g_{2i} := x_i \wedge z_{i-1};\ u_i := u_{i-1} \vee g_{2i}.$$

Заключительные присваивания в схеме

$$g_{2n+1} := \neg u_n;\ g_{2n+2} := z_n \wedge g_{2n+1}.$$

Из построения видно, что в схеме $O(n)$ присваиваний. Докажем, что $g_{2n+2}(x) = U_2(x)$ для любого набора значений переменных x_i. Для этого по индукции докажем, что выполняются указанные свойства переменных y_i, z_i и u_i.

База индукции $i = 0$ очевидно справедлива.

Пусть свойства выполнены при всех $j < i$. Рассмотрим значения переменных y_i, z_i и u_i.

По построению $y_i = 1$ если $x_i = 1$ или среди первых $i - 1$ переменных x_j есть хотя бы одна 1. Но это и означает, что среди первых i переменных x_j есть хотя бы одна 1.

Поскольку $z_i = (x_i \wedge y_{i-1}) \vee z_{i-1}$, то $z_i = 1$ тогда и только тогда, когда среди первых i переменных x_j есть хотя бы две 1 (или это свойство выполнено уже при $i - 1$, или $x_1 = 1$ и среди первых $i - 1$ переменных x_j есть хотя бы одна 1).

Аналогичное рассуждение справедливо и для переменных u_i.

Значение схемы равно 1 тогда и только тогда, когда $z_n = 1$ и $u_n = 0$. Но это и означает, что единиц ровно 2 (не меньше двух и не больше двух).

6. Постройте вычислимую биекцию $f\colon \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^*$ между множеством пар двоичных слов и множеством двоичных слов. Укажите явно правило, которое сопоставляет паре слов значение этой биекции. Найдите для построенной биекции (а) $f(00, 11)$, (б) $f^{-1}(10)$.

Решение. Определим функцию удвоения символов: если $u = u_1u_2 \dots u_n$, $u_i \in \{0,1\}$, то

$$D(u) = u_1u_1u_2u_2 \dots u_nu_n.$$

Легко проверить индукцией по длине слова, что D инъективная.

Теперь определим искомую биекцию следующими правилами (λ — пустое слово):

$$f(u,v) = \begin{cases} D(u)10v, & \text{если } (u \text{ непустое}) \text{ или } (u = \lambda,\ v = (10)^k D(x)10y \text{ для некоторых слов } x,\ y); \\ v, & \text{в противном случае.} \end{cases}$$

Докажем, что отображение $g(u,v) = D(u)10v$ — инъекция из $\{0,1\}^* \times \{0,1\}^*$ в $\{0,1\}^*$. Пусть

$$g(u,v) = D(u)10v = D(x)10y = g(x,y).$$

Кратчайший префикс, заканчивающийся на 10, в этом слове — это $D(u) = D(x)$. Поэтому $u = x$, так как D инъективная. А поскольку совпадают суффиксы после первого вхождения 10, то $v = y$.

Очевидно, что функция $h(w) = (\lambda, w)$ является инъекцией из $\{0,1\}^*$ в $\{0,1\}^* \times \{0,1\}^*$.

Построенная выше f получается из доказательства теоремы Кантора–Бернштейна применительно к этим двум инъекциям. В данном случае мы получаем полубесконечные цепочки двух видов.

Цепочки первого вида начинаются с пары (u,v), где u непустое или $u = \lambda$, $v = (10)^k D(x)10y$ для некоторых слов x, y. Они выглядят как

$$(u,v) \xrightarrow{g} D(u)10v \xrightarrow{h} (\lambda, D(u)10v) \xrightarrow{g} 10D(u)10v \xrightarrow{h} (\lambda, 10D(u)10v) \xrightarrow{g} \dots \xrightarrow{g} (10)^k D(u)10v \xrightarrow{h} \dots$$

Им отвечает первое правило в определении f.

Цепочки второго вида начинаются со слова v, где v не лежит в образе g, то есть не представляется в виде $D(x)10y$. Они выглядят как

$$v \xrightarrow{h} (\lambda, v) \xrightarrow{g} 10v \xrightarrow{h} (\lambda, 10v) \xrightarrow{g} (10)^2 v \xrightarrow{h} (\lambda, (10)^2 v) \xrightarrow{g} \dots \xrightarrow{g} (10)^k v \xrightarrow{h} \dots$$

Им отвечает второе правило в определении f: для построения биекции $\{0,1\}^* \times \{0,1\}^* \to \{0,1\}^*$ на этих цепочках нужно идти «против» стрелок h.

(а) По определению, $f(00,11) = 00001011$, так как 00 непустое.

(б) Слово 10 принадлежит цепочке, начинающейся со слова λ:

$$\lambda \xrightarrow{h} (\lambda, \lambda) \xrightarrow{g} 10 \xrightarrow{h} (\lambda, 10) \xrightarrow{g} \dots$$

Поэтому $f^{-1}(10) = (\lambda, 10)$.

7. Известно, что функция $f\colon \mathbb{N} \to \mathbb{N}$ невычислима. Верно ли, что множество её значений

$$f(\mathbb{N}) = \{y : y = f(x) \text{ для некоторого } x\}$$

неразрешимо?

Решение. Неверно.

Функций $\mathbb{N} \to \{0,1\}$ (то есть бесконечных двоичных последовательностей) континуум. Значит, среди них есть невычислимые. Но множество $\{0,1\}$ конечно и потому разрешимо.

8. Пусть $U(p,x)$ — главная универсальная вычислимая функция. Докажите, что перечислимо множество тех программ, которые вычисляют функцию, не равную 0 на всей области определения. (Формально это множество таких чисел p, что $U(p,x) \neq 0$ для некоторого x.)

Замечание. Утверждение задачи неверно. Дело в том, что условие $U(p,x) \neq 0$ выполняется и в том случае, если $U(p,x)$ не определена. В частности, в множество программ входят все программы, вычисляющие функции, которые не определены хотя бы в одной точке.

Доказательство неперечислимости указанного множества программ довольно трудное. Его можно получить из критерия, описывающего все перечислимые свойства вычислимых функций. Формулировку критерия см. в книге Н.К.Верещагин, А.Шень «Вычислимые функции», теорема 24. Проверка того, что указанное в задаче множество программ неперечислимо, основывается на этом критерии.

Подразумевавшееся условие задачи 8. Пусть $U(p,x)$ — универсальная вычислимая функция. Докажите, что перечислимо множество тех программ, которые вычисляют функцию, не равную 0 на всей области определения, причем область определения непуста. Формально это множество таких чисел p, что $U(p,x) \neq 0$ для некоторого x и функция U определена на паре (p,x).

Решение. Перечисляющий алгоритм запускает процесс параллельного вычисления всех вычислимых функций на всех входах с попутной проверкой выполнения требуемого свойства функции. Этот процесс устроен следующим образом. Занумеруем пары (p,x) какой-нибудь вычислимой биекцией $c\colon \mathbb{N} \times \mathbb{N} \to \mathbb{N}$. Обратную биекцию обозначим $n \mapsto (\pi_1(n), \pi_2(n))$.

Параллельное вычисление всех функций на всех входах происходит по стадиям: первая стадия состоит в исполнении первого такта алгоритма $\pi_1(1)$ на входе $\pi_2(1)$, вторая — второго такта алгоритма $\pi_1(1)$ на входе $\pi_2(1)$ и первого такта алгоритма $\pi_1(2)$ на входе $\pi_2(2)$ и т.д.: на N-й стадии исполняется N-й такт алгоритма $\pi_1(1)$ на входе $\pi_2(1)$, $(N-1)$-й такт алгоритма $\pi_1(2)$ на входе $\pi_2(2)$, ..., первый такт алгоритма $\pi_1(N)$ на входе $\pi_2(N)$.

Если алгоритм p на входе x остановился, то на дальнейших стадиях эта пара пропускается.

Как уже говорилось, при исполнении каждого такта также выполняется проверка. Если алгоритм p на входе x остановился с результатом $r \neq 0$, то перечисляющий алгоритм печатает p.

Из описания алгоритма видно, что будут перечислены только номера функций, которые принимают значение, отличное от 0.

С другой стороны, из описания процесса параллельного исполнения ясно, что каждый такт работы алгоритма p на входе x будет исполнен на некоторой стадии. Поэтому если $U(p,x) \neq 0$ и $U(p,x)$ определена, это будет обнаружено и номер p будет напечатан. Это завершает доказательство корректности алгоритма перечисления.

Второе решение. Запишем то же самое рассуждение с помощью отладочной функции. Нам потребуется вычислимая биекция $\tau\colon \mathbb{N}^3 \to \mathbb{N}$, которую легко изготовить из вычислимой биекции $c\colon \mathbb{N} \times \mathbb{N} \to \mathbb{N}$:

$$\tau(x,y,z) = c(x, c(y,z)).$$

Обратная биекция имеет вид $\tau^{-1}(n) = (\pi_1(n), \pi_1(\pi_2(n)), \pi_2(\pi_2(n)))$.

Перечисляющий алгоритм перебирает все натуральные числа в порядке возрастания, для каждого числа n вычисляет $\tau^{-1}(n) = (p, x, t)$; затем вычисляет отладочную функцию $F(p, x, t)$, если $F(p, x, t) = 1$ и $U(p, x) \neq 0$, то перечисляющий алгоритм печатает p. Доказательство корректности то же самое.