АМВ. ДЗ на неделю 6.

Прохоров Юрий, 771

Задача 1

Докажите, что $\mathcal{RP} \subset \mathcal{NP}$.

Решение:
Будем пользоваться определением из конспекта и считать, что класс $\mathcal{RP}$ состоит из языков L, для которых существует полиномиальная машина Тюринга M такая, что

$$\mathbb{P}_r\{M(x,r) = 1 \mid x \notin L\} = 0$$

$$\mathbb{P}_r\{M(x,r) = 1 \mid x \in L\} \geq \frac{1}{2}$$

Так как M работает полиномиально, то за время работы она успеет прочитать только полиномиальное число случайных битов.

Если $x \in L$, то вероятность принятия x машиной M ненулевая и такая полиномиальная последовательность битов существует, и ее можно взять в качестве сертификата. Если $x \notin L$, то такой последовательности не существует, так как вероятность принятия нулевая.

Задача 2

Покажите, что в задаче сравнения больших файлов, разобранной на семинаре, вероятность ошибки действительно не превосходит $3/4$ при достаточно больших n. Оцените, насколько должно быть велико n и покажите, что n бит ≥ 32 мегабайта — достаточное количество для справедливости оценок.

Решение:

Пусть $|x| = |y| = n$ — поданные на вход слова (файлы, числа в двоичной записи).

Алгоритм (сравнения файлов):

1. Выбрать случайное простое число p из промежутка $[n, 2n]$.
2. Вычислить
$$x \equiv u \pmod{p}, \qquad y \equiv v \pmod{p}$$
3. Сравнить числа u и v как двоичные строки.

Если $x = y$, то алгоритм работает корректно всегда. Будем искать вероятность ошибки алгоритма, при условии, что поданные на вход слова x и y не равны.

Ошибка происходит тогда и только тогда, когда

$$p \mid |x - y|, \qquad x \neq y.$$

Пусть $\pi(n)$ — число простых чисел, меньших n. Пусть число $|x-y|$ имеет k различных простых делителей $p_1, \ldots, p_k$ на промежутке $[n, 2n]$. Тогда ошибка будет, если p совпадет с одним из них:

$$\mathbb{P}\{error \mid x \neq y\} = \frac{k}{\pi(2n) - \pi(n)}$$

Справедливы оценки:

$$2^n \geq |x-y| \geq p_1 p_2 \cdot \ldots \cdot p_k \geq n^k \qquad \Longrightarrow \qquad k \leq \frac{n \ln 2}{\ln n}$$

$$\mathbb{P}\{error \mid x \neq y\} \leq \frac{n \ln 2}{\ln n} \frac{1}{\pi(2n) - \pi(n)}$$

Известно, что

$$\lim_{n\to\infty} \frac{\pi(n) \ln n}{n} = 1 \qquad \Longleftrightarrow \qquad \forall \varepsilon > 0 \ \exists n_0 : \ \forall n > n_0 \ :$$

$$(1-\varepsilon)\frac{n}{\ln n} \leq \pi(n) \leq (1+\varepsilon)\frac{n}{\ln n}$$

Тогда после некоторых преобразований можно получить оценку:

$$\mathbb{P}\{error \mid x \neq y\} \leq \frac{\ln 2}{(2-2\varepsilon)\frac{1}{1+\frac{\ln 2}{\ln n}} - (1+\varepsilon)} \leq \frac{3}{4}, \qquad \text{при } n \geq n_1$$

Отсюда

$$\varepsilon \leq \frac{3 - 4\ln 2}{9} \approx 0.0253, \qquad \log_2 n \geq \frac{3 + 4\ln 2 + 3\varepsilon}{3 - 4\ln 2 - 9\varepsilon}$$

Согласно странице «Prime counting function» Википедии, оценка $\varepsilon \leq 0.02$ выполняется примерно при $n_0 = 10^{18} \approx 2^{60}$. При этом $\log_2 n \geq 63$. Поэтому данная оценка является очень грубой.

Задача 3

Покажите, что класс $\mathcal{BPP}$ не изменится, если

(i) константу стандарта Монте-Карло $\frac{1}{3}$ заменить на любое число, строго меньшее $\frac{1}{2}$.

(ii) полиномиальное *в среднем* число шагов заменить на просто полиномиальное число шагов.

Решение:

(i) Пусть на вход подается слово x, и (условная) вероятность ошибки на этом слове

$$\mathbb{P}_0\{\text{error} \mid x\} \leq p < \frac{1}{2}$$

Построим следующий алгоритм. Запустим данный базовый алгоритм $(2n+1)$ раз и выведем ответ, который встретился больше. В таком случае новый алгоритм выдаст ошибку тогда и только тогда, когда базовый алгоритм ошибется $n+1$ или более раз.

$$\mathbb{P}_0\{k \text{ errors out of } 2n+1 \text{ trials} \mid x\} = C_{2n+1}^k p^k (1-p)^{2n+1-k}$$

$$\mathbb{P}\{\text{error} \mid x\} = \sum_{k=n+1}^{2n+1} \mathbb{P}_0\{k \text{ errors} \mid x\} = \sum_{k=0}^{n} C_{2n+1}^k (1-p)^k p^{2n+1-k}$$

Пусть

$$f(x) = (1-p)^x p^{2n+1-x}, \qquad 0 < p < \frac{1}{2}$$

$$f'(x) = f(x)\big[\ln(1-p) - \ln p\big] = f(x) \ln \frac{1-p}{p} > 0$$

В связи с этим, можно сделать оценку сверху при $k = n$ в сумме:

$$\mathbb{P}\{\text{error} \mid x\} \leq (1-p)^n p^{n+1} \sum_{k=0}^{n} C_{2n+1}^k = \frac{p}{2}(1-p)^n p^n 2^{2n+1} = p\big[4p(1-p)\big]^n = p\big[1-(2p-1)^2\big]^n \longrightarrow 0, \ n \to \infty$$

Это верно только при $p < \frac{1}{2}$. Для фиксированного p существует константа n_0 такая, что после повторения алгоритма $2n_0 + 1$ раз вероятность ошибки на слове x будет меньше $\frac{1}{3}$.

Полиномиальность не нарушена, так как n_0 — фиксированная константа для данного p.

(ii) Пусть вероятностная машина принимает слова по стандарту Монте-Карло, то есть ошибается с вероятностью менее $\frac{1}{3}$. Пусть $P(n)$ — полином, которым в среднем ограничено время работы некоторого алгоритма. Пусть $\forall n$ существует вход длины n, на котором алгоритм работает дольше любого полинома от длины входа.

Тогда существует хотя бы фиксированная доля входов $\alpha > 0$, на которых машина работает строго меньше $P(n)$, иначе бы полиномиальная оценка в среднем не была бы возможна. Пусть t — время работы алгоритма. Тогда

$$\mathbb{P}\{t < P(n)\} \geq \alpha > 0$$

Построим следующий алгоритм: будем запускать базовую машину, но как только она достигает времени работы $t = P(n)$, будем выводить случайный ответ и завершать ее работу. В случае, если выдается случайный ответ, вероятность ошибки будет $\frac{1}{2}$. Итак, вероятность ошибки нового алгоритма:

$$\mathbb{P}\{\text{error}\} = \mathbb{P}\{\text{error} \mid t < P(n)\} \cdot \mathbb{P}\{t < P(n)\} + \mathbb{P}\{\text{error} \mid t \geq P(n)\} \cdot \mathbb{P}\{t \geq P(n)\}$$

$$\mathbb{P}\{\text{error}\} \leq \frac{1}{3}\alpha + \frac{1}{2}(1-\alpha) = \frac{1}{2} - \frac{\alpha}{6} < \frac{1}{2}$$

Таким образом, мы свели задачу к первому пункту.

Задача 4

Проверьте матричное равенство $C = AB$, где A, B, C — $n \times n$ матрицы, имеющие целочисленные элементы, не превосходящие по абсолютной величине число h, используя рандомизацию.

(i) Каким нужно выбрать число N, чтобы вероятность ошибки процедуры была меньше заданной вероятности p?

(ii) Для дальнейшей экономии вы решили использовать проверку

$$x^T ABx = x^T Cx \qquad \text{или} \qquad y^T ABx = y^T Cx,$$

где вектор y выбирается независимо от x. Как изменится N для этих случаев?

Решение:
(i) Алгоритм:

1. Выбрать случайный вектор x из целых чисел от 0 до $N-1$.
2. Вычислить Cx и ABx и сравнить значения.

Если $C = AB$, то алгоритм всегда будет давать верный ответ. Найдем вероятность, что алгоритм ошибается, при условии, что $C \neq AB$.

Пусть $x = (x_1, \ldots, x_n)^T$ — вектор из независимых переменных. Из линейной алгебры известно, что

$$C = AB \qquad \Longleftrightarrow \qquad \forall x: \; Cx = ABx,$$

так как в качестве x можно брать векторы стандартного базиса. Пусть

$$Cx = \big(P_1(x), \ldots, P_n(x)\big), \qquad ABx = \big(Q_1(x), \ldots, Q_n(x)\big),$$

где $P_i(x), \; Q_j(x)$ — многочлены первой степени от переменных $x_1, \ldots, x_n$.

$$C = AB \qquad \Longleftrightarrow \qquad P_i(x) \equiv Q_i(x), \qquad i = 1, \ldots, n \qquad \Longleftrightarrow$$

$$\Longleftrightarrow \qquad R(x) = \Big(P_1(x) - Q_1(x)\Big)^2 + \ldots + \Big(P_n(x) - Q_n(x)\Big)^2 \equiv 0, \qquad \deg R(x) = 2$$

Известно, что на вход подается $C \neq AB$, поэтому $R(x) \not\equiv 0$.

Лемма Шварца-Зиппеля
Пусть

- $R(x_1, \ldots, x_n)$ — *многочлен над полем* $\mathbb{F}$,
- $\deg R(x) = d$ — *максимальная степень по каждой переменной*,
- $R(x) \not\equiv 0$,
- $S \subseteq \mathbb{F}$ — *конечное подмножество* $\mathbb{F}$.

Если $r_1, \ldots, r_n$ — *случайные значения из* S, *то вероятность*

$$\mathbb{P}\{R(r_1 \ldots, r_n) = 0\} \leq \frac{dn}{|S|}.$$

По лемме Шварца-Зиппеля:

$$\mathbb{P}\{\text{error} \mid C \neq AB\} = \mathbb{P}\{R(x) = 0\} \leq \frac{dn}{|S|} = \frac{2n}{N} < p \qquad \Longrightarrow \qquad N > \frac{2n}{p}$$

(*ii*) В первом случае аналогичный алгоритм, но теперь вычисляется значения многочлена

$$R(x) = x^T ABx - x^T Cx = x^T (AB - C)x,$$

x который является некоторой билинейной формой, поэтому $\deg R(x) = 2$.

По лемме Шварца-Зиппеля:

$$\mathbb{P}\{\text{error} \mid C \neq AB\} = \mathbb{P}\{R(x) = 0\} \leq \frac{2n}{N} < p \qquad \Longrightarrow \qquad N > \frac{2n}{p}$$

Во втором случае вычисляется многочлен

$$R(x, y) = y^T (AB - C)x, \qquad \deg R(x, y) = 2$$

Аналогично, по лемме Шварца-Зиппеля:

$$\mathbb{P}\{\text{error} \mid C \neq AB\} = \mathbb{P}\{R(x) = 0\} \leq \frac{2 \cdot 2n}{N} < p \qquad \Longrightarrow \qquad N > \frac{4n}{p}$$

Задача 5

(*i*) Покажите, что вероятность того, что случайно выбранное ребро в графе входит в минимальный разрез не превышает $\frac{2}{|V|}$.

(*ii*) Покажите, что алгоритм $MINCUT$ выдает минимальный разрез с вероятностью $\geq \frac{2}{n(n-1)}$, где $|V| = n$.

(*iii*) Покажите, что если независимо повторить процедуру $MINCUT$ n^2 раз, то минимальный разрез будет найдет с вероятностью > 0.85.

Решение:

Лемма. *Число ребер в минимальном разрезе графа* G:

$$C \leq \frac{2|E|}{|V|} = \frac{2m}{n}.$$

Доказательство.
Пусть C — вес минимального разреза (число ребер в нем). Пусть существует вершина степени меньше C. Тогда эту вершину можно поместить в отдельное множество и вес разреза уменьшится — противоречие. Значит, по лемме о рукопожатиях:

$$2m = \sum_{v \in V} \deg(v) \geq Cn \qquad \Longleftrightarrow \qquad C \leq \frac{2m}{n}.$$

Лемма доказана. □

(i) Пусть C — вес минимального размера.

$$\mathbb{P}\{e \in \min \text{ cut}\} = \frac{C}{m} \leq \frac{2}{n}$$

(ii) Алгоритм $MINCUT$:

1. Выбрать случайное ребро графа G и стянуть его.
2. Повторить шаг 1, пока не останется мультиграф на двух вершинах.

Вершины, стянутые в разные вершины, лежат в разных разрезах. Таким образом, минимальный разрез будет найден тогда и только тогда, когда ни одного ребра это разреза не будет стянуто. С учетом пункта 1 найдем вероятность правильного ответа:

$$\mathbb{P}\{\text{correct}\} \geq \big(1 - \frac{2}{n}\big)\big(1 - \frac{2}{n-1}\big) \dots \big(1 - \frac{2}{3}\big) = \frac{2}{n(n-1)}$$

(iii) Повторим алгоритм n^2 раз и выберем из всех найденных разрезов максимальный. Ошибка будет, если во все разы алгоритм не смог найти минимальный разрез. Вероятность неправильного ответа:

$$\mathbb{P}\{\text{wrong}\} = \Big(1 - \frac{2}{n(n-1)}\Big)^{n^2}$$

$$\lim_{n\to\infty} \Big(1 - \frac{2}{n(n-1)}\Big)^{n^2} = \lim_{x\to\infty} \Big(1 + \frac{1}{\frac{x(1-x)}{2}}\Big)^{\frac{x(1-x)}{2}\frac{x^2\cdot 2}{x(1-x)}} = e^{-2} = \frac{1}{e^2} \approx 0.135$$

Поэтому при достаточно больших n:

$$\mathbb{P}\{\text{correct}\} > 0.865 > 0.85$$

Задача 6

Докажите, что 2-$CNF \in \mathcal{P}$.

Решение:
Пусть на вход подается формула $\phi = C_1 \wedge \ldots \wedge C_m$ от переменных $x_1, \ldots, x_n$, где $C_i = (a_i \vee b_i)$.

Алгоритм проверки на выполнимость:

1. Заменить каждый $C_i = (a_i)$ на $C_i = (a_i \vee a_i)$.
2. Заменить каждый $C_i = (a_i \vee b_i)$ на эквивалентную формулу $C_i = (\neg a_i \to b_i) \wedge (\neg b_i \to a_i)$.
3. Построить граф G с $2n$ вершинами, соответствующими литералам $x_i,\ \neg x_i$ $(i = 1, \ldots, n)$.
4. Для каждой импликации $(c_i \to d_i)$ в ϕ провести ориентированное ребро $[c_i, d_i]$ в графе G.
5. Провести конденсацию графа с помощью поиска в глубину.
6. Если в одной компоненте сильной связности есть пара вершин x_i и $\neg x_i$ для какого-либо i, то КНФ невыполнима, иначе — выполнима.

Все действия требуют полиномиального времени.

При доказательстве корректности будем использовать следующее утверждение:

$$(a \longrightarrow b) = 1 \qquad \Longleftrightarrow \qquad a \leq b$$

Также сформулируем следующее свойство графа: пусть из вершины a достижима вершина b. Тогда из $\neg b$ достижима вершина $\neg a$. Это следует из того, что любому ребру в G соответствует другое инвертированное ребро, концы которого есть отрицания концов первого.

• Пусть 2-КНФ ϕ выполнима. Допустим, в одной компоненте сильной связности есть, для определенности, вершины x_1 и $\neg x_1$. Тогда в ориентированном графе G есть путь из x_1 в $\neg x_1$:

$$x_1 \to a_{i_1} \to \ldots \to a_{i_k} \to \neg x_1$$

По построению графа, в ϕ должны быть импликации $(x_1 \to a_{i_1}), \ldots, (a_{i_k} \to \neg x_1)$. На выполняющем наборе они все должны обратиться в 1, поэтому с учетом утверждения выше:

$$x_1 \le a_{i_1} \le \ldots \le a_{i_k} \le \neg x_1 \qquad \Longrightarrow \qquad x_1 \le \neg x_1$$

Аналогично, есть путь из $\neg x_1$ в x_1, поэтому $\neg x_1 \le x_1$. Тогда $x_1 = \neg x_1$ при выполняющем наборе — противоречие.

• Пусть литералы в каждой компоненте сильной связности графа G независимы ($\forall i\ x_i, \neg x_i$ лежат в разных компонентах). Построим выполняющий набор. В одной компоненте сильной связности любая вершина достижима из любой другой, поэтому в силу рассуждений, аналогичных доказательству в другую сторону, литералы из одной компоненты сильной связности должны принимать одно значение.

Нам нужно занумеровать все вершины графа так, чтобы x_i и $\neg x_i$ имели разные значения, и из вершин со значением 1 не были достижимы вершины со значением 0, то есть ребра шли только в направлении неубывания значений.

Пусть H — конденсат графа. Тогда H — ориентированный, ациклический граф, не обязательно связный. Задача сводится к нумерации вершин графа H так, чтобы ребра шли только в направлении неубывания.

Отсортируем топологически каждую компоненту H. Возьмем любую компоненту и ее первую (в порядке сортировки) мультивершину. Ее занумеруем нулем. Будем продолжать нумеровать все достижимые из нее вершины нулями, пока в исходном графе мы не закрасим вершины a_i и $\neg a_i$ для некоторого i. Компоненту с $\neg a_i$ нумеруем единицей, все достижимые из нее вершины — тоже. Также делаем для остальных компонент связности H, избегая одинаковых нумераций вершин b и $\neg b$.

Пусть в H есть компонента связности, в которой из вершины $a = 1$ достижима $\neg a = 0$. Пусть $b = 0$ и $\neg b = 1$ — вершина, на которой сменилось значение при нумерации. То есть из b достижимо $\neg b$, из $\neg b$ достижимо a, из a достижимо $\neg a$. По свойству выше, из $\neg a$ достижимо $\neg b$. Но тогда a и $\neg a$ — в одной компоненте связности — противоречие.

Задача 7

Пронумеруем карты сверху вниз от 1 до n. Далее в цикле берем карту сверху и вставляем в случайное место. Цикл заканчивается после того, как карта номер $n-1$ была взята сверху и поставлена в случайное место.

Докажите, что:

(i) всё, что под картой номер $n-1$ равномерно перемешано на любом шаге цикла (все перестановки там равновероятны).

(ii) вставка карты случайно и независимо в некоторое место уже равномерно перемешанной колоды генерирует также равномерно перемешанную колоду.

(iii) Найдите матожидание количества итераций цикла.

Решение:
За 1 шаг алгоритма будем считать одно поднятие карты $(n-1)$.

(i) Докажем это утверждение в предположении, что мы знаем, какие именно карты оказались под картой $(n-1)$. Доказательство проведем по индукции.

База: на нулевом шаге (0 поднятий карты $n-1$) под $(n-1)$-ой картой лежит только 1 карта, у которой только одно перестановка.

Предположение: пусть на $(k-1)$-ом шаге ($k \geq 1$) под картой $n-1$ лежат карты $a_1, \ldots, a_{k-1}$, которые равномерно перемешены.

Переход: пусть мы знаем, что сейчас под карту $n-1$ попадет карта a_k. Посчитаем вероятность произвольной перестановки. Карта a_k может попасть на k различных мест, будем считать первым местом — самое верхнее (прямо под картой $n-1$). Позицию, на которую попадает a_k, обозначим $\mathrm{pos}(a_k)$.

Все вероятности будет считать при условии, что под картой $n-1$ лежат именно $a_1, \ldots, a_{k-1}$ и сейчас добавляется карта a_k. По формуле полной вероятности:

$$\mathbb{P}\{(a_{i_1}, \ldots, a_{i_k})\} = \sum_{\phi \in S_{k-1}} \mathbb{P}\{(a_{i_1}, \ldots, a_{i_k}) \mid \phi\} \cdot \mathbb{P}\{\phi\} = \frac{1}{(k-1)!} \sum_{\phi \in S_{k-1}} \mathbb{P}\{(a_{i_1}, \ldots, a_{i_k}) \mid \phi\},$$

где $\phi \in S_{k-1}$ — все перестановки $k-1$ элементов.

Чтобы слагаемое в сумме было отлично от нуля, необходимо и достаточно, чтобы перестановка ϕ была подпоследовательностью данной перестановки $(a_{i_1}, \ldots, a_{i_k})$ и не содержало a_k. У последовательности длины k ровно 1 такая подпоследовательность длины $k-1$, поэтому в сумме будет 1 слагаемое.

$$\mathbb{P}\{(a_{i_1}, \ldots, a_{i_k})\} = \frac{1}{(k-1)!} \cdot \mathbb{P}\{\mathrm{pos}(a_k) = s \mid (a_{i_1}, \ldots, a_{i_{s-1}}, a_{i_{s+1}}, \ldots, a_{i_{k-1}})\} = \frac{1}{(k-1)!} \cdot \frac{1}{k} = \frac{1}{k!}$$

Всего перестановок $k!$, поэтому каждая из них равновероятна.

(*ii*) Пусть колода перемешана равномерно. Найдем вероятность произвольной перестановки после одной случайной вставки верхней карты.

Так как колода равномерно перемешена, то и под верхней картой все $n-1$ карт равномерно перемешаны. Но это как раз есть утверждение индукции из предыдущего пункта при $k = n-1$, которое уже доказано.

(*iii*) Посчитаем матожидание числа итераций цикла, то есть количества вызовов генератора случайных чисел. Пусть ξ — случайная величина, равная этому количеству. Пусть ξ_k — число вызовов генератора на k-ом шаге алгоритма ($1 \leq k \leq n-1$). В силу линейности матожидания:

$$\xi = \sum_{k=1}^{n-1} \xi_k \qquad \Longrightarrow \qquad \mathbb{E}_\xi = \sum_{k=1}^{n-1} \mathbb{E}[\xi_k]$$

Пусть p_m — вероятность вызвать генератор ровно m раз на k-ом шаге алгоритма. На этом шаге под картой $n-1$ лежит k карт, значит, есть $k+1$ «успешная» позиция. Чтобы было ровно m вызовов, нужно $m-1$ раз попасть в «неуспешную» позицию, а на m-ый раз в «успешную».

$$p_m = \Big(1 - \frac{k+1}{n}\Big)^{m-1} \frac{k+1}{n} = \frac{k+1}{n} \cdot \Big(\frac{n-k-1}{n}\Big)^{m-1}$$

$$\mathbb{E}[\xi_k] = \sum_{m=1}^{\infty} m p_m = \frac{k+1}{n} \sum_{m=1}^{\infty} m \Big(\frac{n-k-1}{n}\Big)^{m-1}$$

Известно, что

$$\frac{1}{1-x} = \sum_{m=0}^{\infty} x^m, \qquad |x| < 1$$

$$\frac{d}{dx} \frac{1}{1-x} = \frac{1}{(1-x)^2} = \sum_{m=1}^{\infty} m x^{m-1}, \qquad |x| < 1$$

При $x = \frac{n-k-1}{n} < 1$:

$$\mathbb{E}[\xi_k] = \frac{k+1}{n} \frac{1}{\big(1 - \frac{n-k-1}{n}\big)^2} = \frac{n}{k+1}$$

Тогда искомое матожидание:

$$\mathbb{E}_\xi = \sum_{k=1}^{n-1} \frac{n}{k+1} = n \sum_{k=2}^{n} \frac{1}{k} = \Theta(n \log n)$$

Задача 8.1

Докажите теорему Татта.

Решение:

Теорема Татта.
По графу $G = (V, E)$ с n вершинами составим кососимметричную матрицу T (матрицу Татта) размера $n \times n$ с элементами t_{ij}:

$$t_{ij} = \begin{cases} 0, & \text{ребро } [i,j] \notin E \\ x_{ij}, & \text{ребро } [i,j] \in E,\ i < j \\ -x_{ji}, & \text{ребро } [i,j] \in E,\ i > j \end{cases},$$

где все x_{ij} являются независимыми переменными.

Тогда в G есть совершенное паросочетание тогда и только тогда, когда $\det T \not\equiv 0$ как многочлен.

Доказательство.

(а) Необходимость.

Пусть в G есть совершенное паросочетание, то есть есть непересекающиеся ребра:

$$[a_1, a_2],\ [a_3, a_4],\ \ldots, [a_{n-1}, a_n], \qquad n \text{ четно}$$

Тогда $x_{12} \not\equiv 0,\ x_{34} \not\equiv 0,\ \ldots, x_{(n-1)n} \not\equiv 0$. Пусть S_n — симметрическая группа, т.е. группа перестановок n элементов. Тогда

$$\det T = \sum_{\phi \in S_n} \operatorname{sign}(\phi) \cdot t_{1\phi(1)} t_{2\phi(2)} \ldots t_{n\phi(n)} = \sum_{\phi \in S_n} C_\phi$$

Пусть $\psi \in S_n$ раскладывается в транспозиции:

$$\psi = (a_1\ a_2)(a_3\ a_4)\ldots(a_{n-1}\ a_n), \qquad \operatorname{sign}(\psi) = \frac{n}{2} \bmod 2$$

Тогда соответствующее ψ слагаемое в детерминанте будет равно (тут под x_{ij} следует понимать $x_{a_i a_j}$ для простоты записи):

$$C_\psi = \operatorname{sign}(\psi) \cdot x_{12}(-x_{12})x_{34}(-x_{34})\ldots x_{(n-1)n}(-x_{(n-1)n}) = x_{12}^2 x_{34}^2 \ldots x_{(n-1)n}^2$$

Нужно показать, что в $\det T$ нет такого же слагаемого, но с минусом, чтобы оно сократилось. Из этого будет следовать, что $\det T \not\equiv 0$. Пусть существует такая перестановка ψ^*, что

$$C_{\psi^*} = -C_\psi.$$

Так как C_{ψ^*} должно содержать те же переменные, что и C_ψ, то $\psi(a_1) = a_2,\ \psi(a_2) = a_1, \ldots$ и так далее. Таким образом, $\psi^* = \psi$ — противоречие.

(б) Достаточность.

Пусть $\det T \not\equiv 0$. Тогда существует такая перестановка ϕ, что $C_\phi \not\equiv 0$ и ни с чем не сокращается при раскрытии детерминанта.

Разложим перестановку ϕ в непересекающиеся циклы:

$$\phi = (a_{11}\ a_{12}\ \ldots\ a_{1l_1})(a_{21}\ \ldots\ a_{2l_2})\ldots(a_{s1}\ \ldots\ a_{sl_s})$$

Допустим, что все циклы имеют четную длину:

$$\forall j\ (1 \le j \le s):\ l_j = 0 \bmod 2$$

Так как ϕ отвечает слагаемому $C_\phi \not\equiv 0$, то все $t_{i\phi(i)}$ являются переменными. Значит, в графе G, есть ребра

$$[a_{j1}, a_{j2}],\ [a_{j3}, a_{j4}],\ \ldots,\ [a_{j(l_j-1)}, a_{jl_j}], \qquad 1 \le j \le s, \qquad s - \text{число циклов в } \phi$$

Все l_j четны, значит эти ребра не пересекаются. Все циклы независимы, значит, ребра от разных циклов не пересекаются. Этих ребер ровно в 2 раза меньше вершин, значит, мы смогли построить совершенное паросочетание в графе G.

Допустим, что существует j $(1 \leq j \leq s)$ такое, что j-ый цикл имеет нечетную длину. Эта длина не может быть равна 1, так как по определению $t_{ii} = 0$, а в нашем случае $C_\phi \not\equiv 0$.

Построим перестановку ϕ^* такую, что она совпадает с ϕ, но j-ый цикл в ней обходится в другую сторону. Длины независимых циклов в ней такие же, значит,

$$\operatorname{sign}(\phi) = \operatorname{sign}(\phi^*).$$

Все переменные в C_{ϕ^*}, которые происходят не от j-ого цикла останутся такими же (там перестановка не изменилась), в все переменные от j-го цикла поменяют знак, потому что матрица T кососимметрична. Так как в j-ом цикле нечетное число элементов, то в C_{ϕ^*} нечетное число переменных поменяет знак, значит,

$$C_{\phi^*} = -C_\phi, \qquad \text{т.к. } \operatorname{sign}(\phi) = \operatorname{sign}(\psi^*)$$

Но это означает, что при раскрытии $\det T$ слагаемое C_ϕ сократится со слагаемым C_{ϕ^*}, что противоречит нашему предположению. Значит, ϕ не может иметь циклы нечетной длины.

Задача 8.2

Задача Д-1 из файла.

Решение:

(i), (ii) Оценим время работы процедуры СТЯГИВАНИЕ(n, k). Она выполняет стягивание ребер $n - k$ ребер. При каждом стягивании надо обработать все ребра, примыкающие к стягиваемым вершинам, на что уйдет время $O(n)$. Итак, время этой процедуры есть $O(n(n-k))$. В алгоритме только вызывается СТЯГИВАНИЕ$\left(n, \frac{n}{2}\right)$, на что уйдет время $O(n^2)$.

Составим рекурренту для асимптотики алгоритма:

$$T(n) = 4T\left(\frac{n}{2}\right) + O(n^2)$$

По мастер-теореме:

$$T(n) = \Theta(n^2 \log n)$$

(iii) Пусть $\mathbb{P}(n)$ — вероятность правильного ответа для графа с n вершинами. Тогда

$$\mathbb{P}(n) = 1 - \left(1 - \frac{1}{4}\mathbb{P}\left(\frac{n}{2}\right)\right)^4 \geq \mathbb{P}\left(\frac{n}{2}\right) - \frac{3}{8}\mathbb{P}\left(\frac{n}{2}\right)^2$$