Задача 1. 1. (2 балла) Да Нет Пропускную способность каждого ребра потоковой сети увеличили на единицу. Верно ли, что максимальный поток также увеличится на единицу.

Задача 1. 2. (2 балла) Да Нет В связном графе из вершины v запустили вначале обход в ширину, а затем обход в глубину. Оказалось, что деревья этих обходов полностью совпали. Значит ли это, что исходный граф был деревом?

Задача 1. 3. (2 балла) Да Нет Во взвешенном неориентированном графе существуют два остова минимального веса. Верно ли, что веса каких-то двух рёбер совпадают?

Задача 1. 4. (2 балла) Да Нет Является ли следующий язык NP-полным? Язык состоит из описаний КНФ, для которых существует как набор значений переменных, обращающий КНФ в истину, так и набор переменных, обращающий КНФ в ложь.

Задача 2. Рассмотрим систему линейных уравнений, записанную в матричной форме:

$$\begin{pmatrix} c_0 & c_3 & c_2 & c_1 \\ c_1 & c_0 & c_3 & c_2 \\ c_2 & c_1 & c_0 & c_3 \\ c_3 & c_2 & c_1 & c_0 \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{pmatrix}$$

Задача 2. 1. (3 балла) Обозначим через $DFT(a_0, a_1, a_2, a_3)$ вектор, равный результату вычисления ДПФ от многочлена $a_0 + a_1x + a_2x^2 + a_3x^3$ (вычисления проводятся в поле комплексных чисел, используя первообразный корня $\omega = i$).

Докажите формулу: если покомпонентно умножить вектор $DFT(c_0, c_1, c_2, c_3)$ на вектор $DFT(x_0, x_1, x_2, x_3)$, то получится вектор $DFT(b_0, b_1, b_2, b_3)$.

Задача 2. 2. (3 балла) Используя результат предыдущего пункта (только такой способ будет оцениваться) решите систему, если $c_0 = 1, c_1 = 2, c_2 = 0, c_3 = 3, b_0 = 1, b_1 = 1, b_2 = 1, b_3 = 1$.

Задача 3.

Неориентированный граф $G(V, E)$ считается связным.

Рассмотрим следующий алгоритм.

(i) Проводим DFS на G, стартуя из произвольной вершины (корня) r. При этом все ребра дерева DFS ориентируем *к корню* r, а все обратные ребра (т. е. ребра графа, не входящие в дерево поиска) — от корня. Получим орграф $\vec{G}$. По построению, любое обратное ребро e будет входить в единственный контур $cycle(e) \in \vec{G}$. Вершинам $\vec{G}$ присвоим стандартные временные DFS-пометки $d(v)$.

(ii) Пометим все вершины $\vec{G}$ как *непросмотренные*.

(iii) Выбираем вершины G,в порядке возрастания их временных пометок. Пусть выбрана вершина v. Если она не просмотрена, то объявляем ее просмотренной.

Обозначим $e_1, \ldots, e_k$ — обратные ребра, исходящие из нее. Для каждого e_i начнем обходить контур $cycle(e_i)$, начиная с e_i, помечая вершины обхода как просмотренные, пока не достигнем просмотренной вершины. Назовем этот обход **цепью**. Ясно, что любая цепь является либо простым путем, либо контуром. Будем представлять цепь списком входящих в нее ребер.

(iv) По выходу из цикла получаем список цепей $C = \{C_1, \ldots, C_p\}$ в порядке их образования. В частности, цепь C_1, по конструкции, всегда является контуром. Назовем список C *цепным разложеним.*

(v) **if** цепное разложения содержит не все ребра, т. е. $\sum_i |C_i| \neq |E|$, **then**

return G *не является* 2*-реберно-связным*

else if в цепном разложении есть цикл, отличный от C_1 **then**

return G 2*-реберно-связный, но не двусвязный*

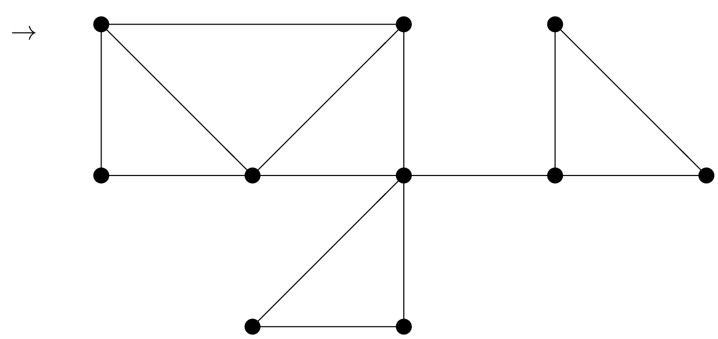
else return G *двусвязный*

В этой задаче нужно обосновать описанный алгоритм. Поскольку процедура выполняется за линейное время, то мы получаем линейный алгоритм определения реберной двухсвязности.

Задача 3. 1. (3 балла) Выразите $p = f(|V|, |E|)$.

Задача 3. 2. (1 балл) Да Нет Верно ли, что цепи не могут иметь общих ребер?

Задача 3. 3. (2 балла) Найдите цепное разложение для графа, изображенного на рисунке.



Задача 3. 4. (5 баллов) Докажите, что ребро графа $e \in E(G)$ является мостом тогда и только тогда, когда оно не входит в цепное разложение.

Задача 3. 5. (5 баллов) Докажите, что вершина $v \in V(G)$ является точкой раздела (точкой сочленения) тогда и только тогда, когда она инцидентна мосту графа или является начальной вершиной какого-то цикла графа, отличного от C_1.

Задача 4. (6 баллов) Пусть $G(V, E)$, — орграф, имеющий m дуг и n вершин, дуги которого помечены целыми числами. Предложите алгоритм трудоемкости $O(n \log n + m)$ для нахождения $s \rightsquigarrow t$-пути с неубывающей последовательностью меток.

Задача 5. Пусть $a = \begin{vmatrix} -t & t \\ 0 & 1 \end{vmatrix}$, $b = \begin{vmatrix} 1 & 0 \\ 1 & -t \end{vmatrix}$ — полиномиальные матрицы. Любое слово w в алфавите $\Sigma = \{a, b\}^*$ определяет матрицу $R(w)$, если конкатенацию символов a и b понимать как произведение соответствующих матриц. Назовем слова u и v в алфавите Σ *эквивалентными*. если $R(u) = R(v)$.

Задача 5. 1. (6 баллов) Постройте эффективный алгоритм, использу.щий вероятностные методы проверки полиномиальных тождеств, устанавливающий эквивалентность (тождество) слов.

Задача 5. 2. (4 балла) Проведите три итерации вашей процедуры проверки эквивалентности слов $u = b^4a^5babab$ и $(ab)^2ab^5a$.

Задача 6. Определим жадный алгоритм для поиска максимального по весу независимого множества в графе G с натуральными весами вершин.

(i) $S \leftarrow \varnothing$; $Gr \leftarrow G$;;

(ii) **while** $Gr \neq \varnothing$

$v \leftarrow$ вершина макс. веса в Gr

$S \leftarrow S \cup v$; $Gr \leftarrow Gr \setminus v$

end while

(iii) **return**(S);

Пусть G — это $n \times n$ грид (решетка).

Задача 6. 1. (5 баллов) Докажите следующее утверждение. Пусть жадный алгоритм выбрал последовательность вершин $v_1, v_2, \ldots, v_m$ (именно в этом порядке), а максимальное независимое множество назовём D. Обозначим $N(v)$ — окрестность вершины v в G, т. е. вершину v и всех ее соседей. Обозначим $A_0 = \varnothing$, $A_k = \cup_{i=1}^{k} N(v_i)\, k = 1, \ldots, m$. Определим последовательность множеств: $B_i = \{D \cap N(v_i) \setminus A_{i-1}\}$, $i = 1, \ldots, m$. Докажите, что сумма весов вершин $w(B_i) \leq 4w(v_i)$.

Пусть G — это $n \times n$ грид (решетка).

Задача 6. 2. (4 балла) Пусть G Покажите. что жадный алгоритм находит независимое множество веса $\frac{OPT}{4}$ (OPT— это вес максимального независимого множества в G).

Задача 7. (4 балла) Да Нет

Верно ли, что следующая задача является NP-полной?

MINIMUM LEAF SPANNING TREE

Вход: неориентированный граф $G(V, E)$, натуральное число k. Верно ли что в G существует остовное дерево, в котором не более k листьев?

Задача 8.

В этой задаче нужно разработать полиномиальный алгоритм для следующей задачи. Дан сильно связный ориентированный граф G. Найти минимальное подмножество дуг в нем, в которых надо сменить ориентацию, чтобы граф стал бы *эйлеровым* , т. е. имел бы циклический обход, включающий каждое ребро ровно один раз.

Задача 8. 1. (2 балла) Докажите, что для разрешимости задачи необходимо, чтобы для любой вершины $v = V(G)$ сумма числа входящих и числа выходящих дуг была четной: $d^+(v) + d^-(v) = 0 \pmod 2$.

Назовем *балансом* вершину v величину $b(v) = \frac{d^+(v)+d^-(v)}{2}$.

Задача 8. 2. (2 балла) Покажите, что если задача разрешима, то необходимо сменить ориентацию не менее $\frac{1}{2}\sum_{v \in V} |bal(v)|$ ребер.

Задача 8. 3. (4 балла) Постройте полиномиальный алгоритм для следующей задачи: найти какое-то подмножество дуг в сильно связном ориентированном графе $G = (V, E)$, ориентацию которых нужно сменить на противоположную, чтобы результирующий граф имел бы эйлеров обход.

Задача 8. 4. (4 балла) Покажите, что исходная задача эквивалентна задаче нахождения потока минимальной стоимости в некоторой сети.

Задача 9. (6 баллов) Разработайте как можно более быстрый алгоритм для следующей задачи

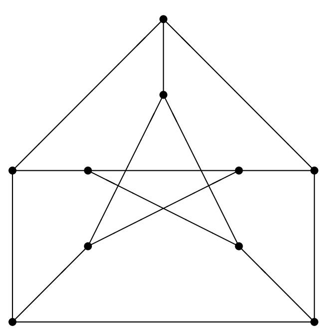
На плоскости заданы n точек. Нужно найти точку (x, y) плоскости (не обязательно из числа заданных), такую что сумма расстояний в метрике ℓ_∞ ($\sum \max(|x - x_i|, |y - y_i|)$) от нее до остальных минимальная.

Задача 10. Пусть дан неориентированный граф $G = (V, E)$, в котором степень каждой вершины равна 3. Пусть $A = (a_{ij})$ – его матрица смежности, т.е.

$$a_{ij} = \begin{cases} 1, & \text{если вершины } i \text{ и } j \text{ соединены ребром,} \\ 0, & \text{иначе.} \end{cases}$$

Определим матрицу $A_r = (a_{ij}^{(r)})$ следующим образом. Пусть $a_{ij}^{(r)}$ – количество путей длины r из вершины i в вершину j в графе G, причём запрещены *возвраты*. Формально: рассматриваются пути $i = x_0 - x_1 - \ldots - x_{r-1} - x_r = j$, где $x_{k-1} \neq x_{k+1}$. Например, $A_1 = A$.

Задача 10. 1. (2 балла) Для графа Петерсена вычислите A_2.



Задача 10. 2. (3 балла) Докажите рекуррентное матричное соотношение $A_{r+1} = A \cdot A_r - 2A_{r-1}$ для $r > 1$.

Задача 10. 3. (1 балл) Корректно доопределите начальное условие $A_0 = ?$, чтобы рекуррента выполнялась и для $r = 1$.

Задача 10. 4. (2 балла) Докажите, что у матрицы A самое большое по модулю собственное значение равно 3.

Задача 10. 5. (4 балла) Можно проверить, что для связного недвудольного графа собственное значение 3 простое (имеет кратность 1), а -3 не является собственным значением. Предполагая эти сведения известными, оцените порядок роста следа матрицы A_r при росте r. А именно, докажите, что $\mathrm{tr} A_r \sim \frac{1}{3} 2^{r+1}$.

Подсказка. Диагонализуйте рекуррентное соотношение ортогональной заменой $S^{-1}AS = diag(\lambda_1, \ldots, \lambda_{|V|})$.