ФИНАЛЬНАЯ КОНТРОЛЬНАЯ. 13.05.2018

ВАРИАНТ 1

1	2	3	4	5	6	7	8	9	10	11	$\sum$

Фамилия, имя студента _____ Группа _____

Фамилия преподавателя, ведущего семинары _____

---

1. Решение каждой задачи должно быть обосновано, ответы без обоснования не принимаются и не оцениваются.
2. В некоторых задачах помимо решения требуется дать краткий ответ “да” или “нет” – это указано в условии после числа баллов за задачу.
3. Можно без доказательства использовать факт **NP**–полноты задач, разобранных на лекциях, на семинарах и описанных в каноническом задании по курсу.
4. Можно без доказательства ссылаться на процедуры, разобранные на лекциях, семинарах или в литературе, при этом обязательно нужно кратко описать, в чём данная процедура заключается.

Простой граф — это граф без петель и кратных ребер.

Задача 1. (2 балла) Дано рекуррентное соотношение:

$$T(n) = \frac{n}{2} \cdot T(n-1) + 1, \quad T(1) = 1$$

Используя **дерево рекурсии**, оцените $T(n)$ как можно точнее. Ваша оценка $f(n)$ должна задаваться явной формулой и должна быть асимптотически эквивалентной $T(n)$, т.е. должна удовлетворять следующему равенству:

$$\lim_{n\to\infty} \frac{f(n)}{T(n)} = 1$$

Решения, не использующие дерево рекурсии, оцениваются из половины баллов.

Решение (А.К.): Расписывая рекурсию в явную формулу, получим:

$$T(n) = 1 + \frac{n}{2} + \frac{n}{2}\cdot\frac{n-1}{2} + \cdots + \frac{n!}{(n-k)!\cdot 2^k} + \cdots + \frac{n!}{2^{n-1}} = \sum_{k=0}^{n-1} \frac{n!}{(n-k)!2^k}$$

$$T(n) = n! \sum_{k=1}^{n} \frac{1}{k!\cdot 2^{n-k}} = \frac{n!}{2^n} \sum_{k=1}^{n} \frac{2^k}{k!} \sim \frac{n!}{2^n}(e^2 - 1)$$

Критерии.

1. Должна быть нарисована понятная картинка дерева рекурсии, чтобы по ней можно было восстановить число уровней и, самое главное, какие веса написаны в вершинах ДР, так чтобы оценка рекурсии сводилась к оценке суммы весов по ДР. (1 балл)

2. Сумма должна быть свернута и должно быть показано, что результат асимптотически эквивалентен $T(n)$. (1 балл)
Если дерева рекурсии нет, то оценивается только второй пункт.

◇

Задача 2. (2+2 балла) В памяти хранится массив чисел $A[1, \ldots, n]$. Назовем **горкой** элемент $A[i]$, который не меньше обоих своих соседей, если $1 < i < n$, или не меньше своего правого или левого соседа, если $i = 1$ или $i = n$.
Чтобы получить полный балл за эту задачу, время работы алгоритма из первого пункта должно соответствовать теоретической нижней оценке, которую нужно получить во втором пункте.

Задача 2. 1. (2 балла) Постройте как можно более быстрый алгоритм, использующий попарные сравнения, находящий "горку" в A, докажите его корректность и оцените число сравнений.

Решение (А.К.): Задача может быть решена за $O(\log n)$ следующей рекурсивной процедурой:

(i) На вход даны l и r – границы рассматриваемого сейчас отрезка. Если l либо r – горка, то возвращаем соответствующий номер. В противном случае $a_l < a_{l+1}$ и $a_{r-1} > a_r$.

(ii) Смотрим на элемент в позиции $m = \lfloor (l+r)/2 \rfloor$. Если он горка, то вызываем его. Иначе либо $a_{m-1} > a_m$, либо $a_m < a_{m+1}$. В первом случае вызываемся рекурсивно к паре (l, m), во втором – к паре (m, r).

(iii) В ходе процедуры в рекурсивных запусках у нас сохраняется инвариант $a_l < a_{l+1}$ и $a_{r-1} > a_r$. Рано или поздно мы либо вернём ответ, либо придём к случаю $r - l \leq 3$. В таком случае либо $l + 1$, либо $r - 1$ будет соответствовать горке. При этом число $r - l$ на каждом шаге уменьшается в два раза, поэтому потребуется $O(\log n)$ шагов.

Критерии.
1. Тривиальный линейный алгоритм оценивается в 0 баллов.

Задача 2. 2. (2 балла) Приведите как можно более точную $\Omega(\cdot)$–оценку числа попарных сравнений, которые должен использовать любой алгоритм, находящий "горку" посредством попарных сравнений.

Решение (А.К.): Рассмотрим решающее дерево, в листьях которого будет номер позиции, в которой найдена горка, а во внутренних вершинах – номера l, r, которые сравниваются. Если $a_l < a_r$, то алгоритм должен переходить в одно поддерево, а иначе – в другое. Любой алгоритм, решающий задачу только попарными сравнениями можно представить в таком виде, при этом высота листа будет равна числу операций, которые нам понадобятся, чтобы найти горку в соответствующей позиции.

Заметим, что для любой позиции k можно привести массив, в котором горка находится ровно в позиции k и нигде больше. Это значит, что в решающем дереве будет хотя бы n листьев, а значит, его высота будет не меньше, чем $\Omega(\log n)$.

Критерии.

1. Тривиальная линейная оценка оценивается в 0 баллов.

Задача 3. (2 балла) Многочлен $A(x) = \sum_{i=0}^{n-1} a_i x^i$ задан последовательностью коэффициентов. Пусть последовательность $\{y_k\}_{k=0}^{n-1}$ — его ДПФ, т. е. $y_k = A\left(e^{\frac{2\pi k}{n} i}\right)$. Предложите алгоритм, вычисляющий $\sum_{k=0}^{n-1}(\mathrm{Re}\ y_k + \mathrm{Im}\ y_k)$ и требующий $o(n^2)$ арифметических операций.

Решение (А.К.): $\sum_{k=0}^{n-1} \mathrm{Re}\ y_k = \mathrm{Re} \sum_{k=0}^{n-1} y_k$, аналогично для Im. Поэтому найдём $\sum_{k=0}^{n-1} y_k$:

$$\sum_{k=0}^{n-1} y_k = \sum_{k=0}^{n-1} \sum_{i=0}^{n-1} a_i \omega_n^{ki} = \sum_{i=0}^{n-1} a_i \sum_{k=0}^{n-1} \omega_n^{ki} = \sum_{i=0}^{n-1} a_i \cdot n \cdot \delta_{i0} = n \cdot a_0$$

Отсюда $\sum_{k=0}^{n-1} \mathrm{Re}\ y_k = n \cdot a_0$, $\sum_{k=0}^{n-1} \mathrm{Im}\ y_k = 0$. Значит, ответ: $n \cdot a_0$.

Критерии.

1.

◇

Задача 4. (1+3+4 балла)

Задача 4. 1. (1 балл) Да Нет
Пусть m — составное число и A — квадратная матрица с элементами из $\mathbb{Z}_m$. Верно ли, что если для некоторого $n > 0$ выполнено $(\det A)^n = 0 \pmod m$, то и $A^n = 0 \pmod m$?
При отрицательном ответе приведите явный контрпример. При положительном ответе — обоснование.

Решение (А.К.): Нет. Пример: $A = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$.

Критерии.
1. Ставится бинарная оценка: решил — не решил.

Задача 4. 2. (3 балла) Пусть p — простое число, A — неотрицательная целочисленная $n \times n$ матрица, элементы которой меньше $m = p^k$, $k > 1$. Предложите алгоритм вычисления $\det A \pmod{p^k}$, использующий $O(n^3 \operatorname{poly}\log m)$ операций.

Решение (А.К.): Будем измерять только число арифметических операций. В конце оно умножится на $\operatorname{poly}\log m$, т.к. мы будем работать только с числами не превышающими $\operatorname{poly} m$ по величине.

Пусть A_i – матрица, составленная из строк и столбцов матрицы A с номерами с i по n *в тот момент времени когда мы её рассматриваем.* Мы будем обращаться к её элементам, подразумевая обращение к A с изменёнными нужным образом индексами.

Пусть $f(i) = \det A_i \pmod{p^k}$. Тогда $\det A \pmod{p^k} = f(1)$. Опишем процедуру, вычисляющую эту функцию.

(i) Если $i = n$, то вернём $a_{nn} \pmod{p^k}$.

(ii) Иначе пусть d – наибольшая степень p, на которую делятся все элементы первого столбца матрицы A_i. Её можно найти за $n \cdot \log m$. Тогда в силу линейности определителя по столбцам мы можем разделить все эти элементы на p^d и получить матрицу A'_i такую что $\det A_i = p^d \det A'_i$. При этом первый столбец матрицы A'_i содержит хотя бы один элемент, который не делится на p.

Поставим строку с этим элементом на первое место, обменяв местами с первой и найдём к нему обратный за $O(\log m)$ с помощью расширенного алгоритма Евклида или двоичного возведения в степень $p^{k-1}(p-1)-1$, после чего вычтем первую строку из остальных с соответствующими коэффициентами, как в методе Гаусса. После этого мы рекурсивно вызовем $f(i+1)$ и вернём $p^d \cdot a_{ii} \cdot f(i+1) \pmod m$, возможно, умножив на -1, если был совершён обмен в начале.

Время работы данной процедуры можно описать таким соотношением:

$$T(i) = T(i+1) + O(n^2),\ T(n) = 1 \implies T(n) = O(n^3)$$

Критерии.
1.

Задача 4. 3. (4 балла) Пусть A — неотрицательная целочисленная $n \times n$ матрица, элементы которой меньше m. Предложите алгоритм вычисления $\det A \pmod m$, использующий $O(n^3 \operatorname{poly} \log m)$ операций.

Решение, опирающееся на факторизацию m оценивается из 2 баллов.

Решение (А.К.): Алгоритм, не требующий факторизации опирается на следующий факт: если у нас есть два числа a и b, то последовательно вычитая меньшее число, умноженное на какой-то коэффициент из большего, мы можем добиться того, что одно из чисел станет равным нулю, а другое – их наибольшему общему делителю.

С помощью этого факта мы можем модифицировать алгоритм Гаусса таким образом: пройдёмся по всем строкам с номерами $j \geq i$ и применим алгоритм Евклида к элементам a_{ii} и a_{ji}. Но при этом с соответствующими коэффициентами мы будем вычитать друг из друга не эти два числа, а строки матрицы.

То есть, пусть в данный момент мы рассматриваем строки i и j. Посмотрим на элементы a_{ii} и a_{ji}. Если $a_{ji} = 0$, то мы уже добились чего хотели, иначе пока это не станет верно будем делать следующую процедуру:

(i) Если $a_{ji} < a_{ii}$, то поменяем i и j строки местами. Определитель умножится на -1, что нужно запомнить.

(ii) Вычтем из строки j строку i с коэффициентом $\lfloor a_{ji}/a_{ii} \rfloor$. При этом все вычисления проводим по модулю m. Из линейной алгебры известно, что такого рода преобразования не будут менять определитель матрицы.

В итоге мы добьёмся того, что в i-ом столбце у всех строк с номерами $j \geq i$ будет стоять нуль. Продолжая данные действия, мы приведём матрицу к верхнетреугольному виду, после чего определитель можно будет посчитать как произведение её диагональных элементов и накопленных -1.

Для каждой пары строк мы проведём $O(\log m)$ итераций алгоритма Евклида, за $O(n)$ каждую, что в сумме даст асимптотику $O(n^3 \log m)$. Данный алгоритм можно ускорить до $O(n^3 + n^2 \log m)$ если находить нужные коэффициенты алгоритмом Евклида над a_{ii} и a_{ji}, а затем заменять i-ю и j-ю строку соответствующими линейными комбинациями.

Критерии.

1.

◇

Задача 5. (5×0.5 баллов)

Пусть L язык, состоящий из (кодировок) пар логических КНФ-формул $\{F_1(\cdot), F_2(\cdot)\}$ от одинакового числа переменных, таких что $F_1 = F_2$. Выберите все нужные варианты ответов на следующие пять вопросов, обоснуйте их и обведите соответствующие поля.
При ответе можно пользоваться стандартными гипотезами:

$$\mathbf{P} \subsetneq \mathbf{NP} \cap \textbf{\textit{co}}\text{–}\mathbf{NP};\ \mathbf{NP} \not\subset \textbf{\textit{co}}\text{–}\mathbf{NP};\ \textbf{\textit{co}}\text{–}\mathbf{NP} \not\subset \mathbf{NP}$$

1. $L \in \mathbf{P}$? Да Нет

Решение (А.К.): Нет. Возьмём $F_2 = 0$, это будет сведение к нашей задаче $\overline{SAT}$. Это значит, что задача **co**–**NP**-трудна.

2. $L \in \mathbf{NP}$? Да Нет

Решение (А.К.): Нет. $\mathbf{NP} \neq$ **co**–**NP**.

3. $L \in \mathbf{NPC}$? Да Нет

Решение (А.К.): Нет. $\mathbf{NP} \neq$ **co**–**NP**.

4. $L \in$ **co**–**NP**? Да Нет

Решение (А.К.): Да. Дополнение данного языка – кодировки пар КНФ-формул таких, что $F_1 \neq F_2$. Оно принадлежит **NP**, т.к. в качестве сертификата можно предложить набор переменных, на которых формулы не равны, а в качестве проверяющего предиката – неравенство F_1 и F_2 на данном наборе.

5. $L \in$ **co**–**NPC**? Да Нет

Решение (А.К.): Да. В силу прошлого пункта и пункта 1.

Критерии.

1. Во всех пунктах ставится бинарная оценка: решил — не решил.

Задача 6. (2 балла) Да Нет

Ребро в потоковой сети называется **критическим**, если уменьшение его пропускной способности уменьшает максимальный поток.

Верно ли, что во всякой потоковой сети, пропускающей ненулевой максимальный поток, найдется хотя бы одно критическое ребро?

При отрицательном ответе постройте явный контрпример. При положительном ответе приведите обоснование.

Решение (А.К./П.О.): Да. Максимальный поток равен минимальному разрезу. Рассмотрим какой-то минимальный разрез. Найдём в нем любое ребро, ведущее из компоненты, содержащей источник, в компоненту, ведущую в сток. Уменьшим пропускную способность этого ребра. Это уменьшит пропускную способность разреза, а значит, и величину максимального потока (в соответствии с теоремой Форда-Фалкерсона). Значит, такой сети не существует.

Критерии.

1.

◇

Задача 7. (2 балла) Да Нет Пусть $G(V, E)$ — простой неориентированный граф, множество вершин которого допускает дизъюнктное разбиение на непересекающиеся подмножества $V = S \sqcup T$, такие, что индуцированные подграфы G_S и G_T являются кликами.

Верно ли, что соответствующий язык всех графов, обладающих таким свойством, принадлежит **NPC**?

По определению, **индуцированный подграф** G_{V_1}, $V_1 \subseteq V(G)$ *имеет вершинами множество* V_1*, а ребрами — все ребра* G *с вершинами из* V_1*.*

Решение (А.К.): Нет. Язык из **P**, т.к. состоит из графов, дополнения которых – двудольные. Данное свойство проверяется за полиномиальное время.

Критерии.

1.

Задача 8. (2+3 балла) Пусть G — ориентированный ациклический граф с неотрицательными весами ребер, в котором выделены вершины s и t, и заданы числа $W_1 \leq W_2$. Вес пути в G равен сумме весов образующих его ребер.

Задача 8. 1. (2 балла) Постройте линейный по входу алгоритм проверки, что вес W максимального пути из s до t попадает в интервал $W_1 \leq W \leq W_2$?
Алгоритм должен быть достаточно подробно описан. Полным баллом оценивается **только линейный алгоритм**

Задача 8. 2. (3 балла) Да Нет
Верно ли, что задача проверки, что вес W *некоторого* пути в G от s до t попадает в интервал $W_1 \leq W \leq W_2$ является NP-полной?

Решение: Первая задача из **P**. Это, по определению, задача о поиске максимального или минимального пути в ДАГ (1 балл), а вот для получения второго балла нужно объяснить, как можно **за линейное время** находить максимальные или минимальные пути во взвешенном DAG'е. Это изложение требует деталей, поскольку в книгах обычно рассмотрен случай без весов. Для решения второй задачи годится старое решение ниже (опушено только прилагательное "наибольший").

Критерии.

1. 1 балл стоит понимание факта, что максимальный по числу ребер путь в ДАГ ищется за линейное время. Оставшийся 1 балла можно получить, если подробно и с деталями изложить, как адаптировать этот алгоритм на случай ребер с весами.

Решение (А.К.): Да. Вспомним постановку задачи о рюкзаке в каноническом задании: Даны числа $a_1, \ldots, a_n$ и число b. Верно ли, что существует поднабор чисел a_i такой что $\sum_i a_i = b$? Построим граф, в котором $n+1$ вершин и из вершины i есть две дуги в вершину $i+1$. Одна из них веса 0, другая – веса a_{i+1}. Тогда если мы найдём путь из 0 в n в таком графе, величина которого равна b (считаем $W_1 = W_2 = b$), мы сможем узнать, можно ли набрать поднабор подходящего веса, т.к. это можно сделать если и только если длина этого пути будет равна b.

Критерии.

1. Кажется, что придется прибегнуть к бинарной оценке: построена сводимость или нет.

◇

Задача 9. (3 балла) Да Нет

Пусть $L = \{(\langle G \rangle, s, t)\}$ — это язык, состоящий из стандартных описаний неориентированных графов G, в которых выделены различные вершины s и t такие, что для любого $S \geq 10$ существует путь из s в t длины S.

Длина пути равна числу рёбер в нем, а в пути допускается повторение вершин и повторение ребер, т. е. можно, например, возвращаться по ребру, по которому только что был сделан переход.

Верно ли, что $L \in$ **co–NP**?

Решение (А.К.): Да, т.к. задача в **P**. Рассмотрим граф G', в котором каждой вершине v исходного графа сопоставлены две вершины v_1 и v_2 нового. Вершине s также будет соответствовать третья вершина – s_0. При этом если в G было ребро (u, v), то в G' должны быть рёбра (u_1, v_2) и (u_2, v_1). А для всех рёбер (s, v) также будут добавлены рёбра (s_0, v_1).

Таким образом любой путь из s_0 в t_2 будет соответствовать пути чётной длины из s в t, а любой путь из s_0 в t_1 будет соответствовать пути нечётной длины из s в t. Это соответствие взаимно однозначное. По условию если есть путь длины $S > 0$, то есть также и путь длины $S + 2$. Значит, нам достаточно найти самый короткий чётный и нечётный пути из s в t и проверить, что оба по величине не больше 10, что может быть сделано с помощью обхода в ширину из s_0 в G'. В этом и только этом случае требуемое условие будет выполняться.

Критерии.

1. Возможно, будут попытки доказать утверждение из определения ***co*–NP**

Задача 10. (3 балла) Дан неориентированный граф G без петель и кратных рёбер, имеющий m рёбер, которым приписаны положительные веса. Раскрасим вершины в два цвета, **трудностью раскраски** назовем наибольший вес ребра между вершинами одного и того же цвета, а если таких рёбер нет, то трудность раскраски считаем равной нулю.

Постройте и обоснуйте $O(m \log m)$–алгоритм, находящий раскраску с **наименьшей трудностью**. *Полиномиальный алгоритм, работающий дольше, чем* $O(m \log m)$*, оценивается из* 1 *балла*

Решение (А.К./Эдуард): Отсортируем рёбра по возрастанию их веса. Наименьшая трудность раскраски (трудность раскраски наименьшей трудности) не больше x если и только если граф, построенный из рёбер, вес которых не меньше x, является двудольным (можем покрасить доли в разные цвета и получить раскраску с трудностью не больше x). В связи с этим можно провести двоичный поиск по величине x и при выборе перехода к (l, m) или (m, r) опираться на то, является ли граф, построенный из рёбер, вес которых не меньше m, двудольным. Если да, то переходим к (l, m), иначе – к (m, r). Итераций двоичного поиска будет не больше $O(\log m)$, на каждой итерации двудольность можно проверить за $O(m)$. Также возможно решение без двоичного поиска с помощью структуры Union-Find.

Критерии.

1.

◇

Задача 11. (2+2 балла) Рассмотрим циркулянтную матрицу порядка $n+1$, первый столбец которой равен $(c_0, c_1, \ldots, c_n)^T$, т. е. матрицу вида

$$\begin{bmatrix} c_0 & c_n & c_{n-1} & \ldots & c_1 \\ c_1 & c_0 & c_n & \ldots & c_2 \\ c_2 & c_1 & c_0 & \ldots & c_3 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ c_n & c_{n-1} & c_{n-2} & \ldots & c_0 \end{bmatrix}$$

Задача 11. 1. (2 балла) Докажите, что все её собственные значения, домноженные на $\frac{1}{\sqrt{n+1}}$, могут быть найдены умножением матрицы Фурье $F_n = \frac{1}{\sqrt{n+1}} \left(\omega_n^{ij}\right)_{i,j=0}^{n}$ размеров $(n+1) \times (n+1)$, где $\omega_n = e^{\frac{2\pi i}{n}}$ — корень из единицы, на вектор $(c_0, c_n, c_{n-1}, \ldots, c_1)^T$.
Указание. Можно без доказательства пользоваться тем фактом, что любая циркулянтная матрица в базисе из столбцов F имеет диагональный вид.

Решение (П. О.): Требуемое сразу следует из того факта, что F диагонализует любой циркулянт: матрица C в базисе из столбцов F представляет собой диагональную матрицу $\Lambda = diag(\lambda_1, \ldots, \lambda_n) = F^{-1}CF$, откуда $C = F\Lambda F^{-1}$, поэтому $CF = F\Lambda$. Далее транспонируем полученное выражение и учтём симметричность (поэлементную) матрицы Фурье: $FC^T = \Lambda F$. Из этого матричного равенства достаточно взять первый столбец левой и правой части.

Критерии.
1.

Задача 11. 2. (2 балла) Найдите с помощью алгоритма БПФ собственные значения циркулянтной матрицы, первый столбец которой имеет вид $(1, 2, 4, 6)^T$.

Решение (П. О.): Используя формулы из предыдущего пункта, находим спектр C: для матрицы, порождённой $(1, 2, 4, 6)^T$ собственные числа $(13, -3+4i, -3, -3-4i)$.

Критерии.
1.

ФИНАЛЬНАЯ КОНТРОЛЬНАЯ. 13.05.2018

ВАРИАНТ 2

1	2	3	4	5	6	7	8	9	10	11	$\sum$

Фамилия, имя студента _____ Группа _____

Фамилия преподавателя, ведущего семинары _____

---

1. Решение каждой задачи должно быть обосновано, ответы без обоснования не принимаются и не оцениваются.
2. В некоторых задачах помимо решения требуется дать краткий ответ “да” или “нет” – это указано в условии после числа баллов за задачу.
3. Можно без доказательства использовать факт **NP**–полноты задач, разобранных на лекциях, на семинарах и описанных в каноническом задании по курсу.
4. Можно без доказательства ссылаться на процедуры, разобранные на лекциях, семинарах или в литературе, при этом обязательно нужно кратко описать, в чём данная процедура заключается.

Простой граф — это граф без петель и кратных ребер.

Задача 1. (2 балла) Дано рекуррентное соотношение:

$$T(0) = 0, \quad T(n) = n \cdot T(n-1) + 1$$

Используя **дерево рекурсии**, оцените $T(n)$ как можно точнее. Ваша оценка $f(n)$ должна задаваться явной формулой и должна быть асимптотически эквивалентной $T(n)$, т.е. должна удовлетворять следующему равенству:

$$\lim_{n\to\infty} \frac{f(n)}{T(n)} = 1$$

Решения, не использующие дерево рекурсии, оцениваются из половины баллов.

Решение (А.К.):

$$T(n) = 1 + n + n(n-1) + n(n-1)(n-2) + \cdots + n! = \sum_{k=1}^{n} \frac{n!}{k!} = n! \cdot \sum_{k=1}^{n} \frac{1}{k!} \sim (e-1) \cdot n!$$

Критерии.
1. Те же, что и в первом варианте.

◇◇

Задача 2. (2+2 балла) В памяти хранится массив чисел $A[1, \ldots, n]$. Назовем **ямкой** элемент $A[i]$, который не больше обоих своих соседей, если $1 < i < n$, или не больше своего правого или левого соседа, если $i = 1$ или $i = n$.
Чтобы получить полный балл за эту задачу, время работы алгоритма из первого пункта должно соответствовать теоретической нижней оценке, которую нужно получить во втором пункте.

Задача 2. 1. (2 балла) Постройте как можно более быстрый алгоритм, использующий попарные сравнения, находящий "ямку" в A, докажите его корректность и оцените число сравнений.

Задача 2. 2. (2 балла) Приведите как можно более точную $\Omega(\cdot)$–оценку числа попарных сравнений, которые должен использовать любой алгоритм, находящий "ямку" посредством попарных сравнений.

Критерии.

1. Те же, что и в первом варианте.

Задача 3. (2 балла) Многочлен $A(x) = \sum_{i=0}^{n-1} a_i x^i$ задан последовательностью коэффициентов. Пусть последовательность $\{y_k\}_{k=0}^{n-1}$ — его ДПФ, т. е. $y_k = A\left(e^{\frac{2\pi k}{n} i}\right)$. Предложите алгоритм, вычисляющий $\sum_{k=0}^{n-1} (\mathrm{Re}\ y_k - \mathrm{Im}\ y_k)$ и требующий $o(n^2)$ арифметических операций.

Критерии.

1. Те же, что и в первом варианте.

◇◇

Задача 4. (1+3+4 балла)

Задача 4. 1. (1 балл) Да Нет
Пусть m — составное число и A — квадратная матрица с элементами из $\mathbb{Z}_m$. Верно ли, что если для некоторого $n > 0$ выполнено $A^n = 0 \pmod m$, то и $(\det A)^n = 0 \pmod m$?

При отрицательном ответе приведите явный контрпример. При положительном ответе — обоснование.

Решение (А.К.): Да. $A^n = 0 \to \det A^n = (\det A)^n = 0$, так как $\det AB = \det A \det B$.
Критерии.
1. Бинарный критерий: решил — не решил.

Задача 4. 2. (3 балла) Пусть p — простое число, A — неотрицательная целочисленная $n \times n$ матрица, элементы которой меньше $m = p^k$, $k > 1$. Предложите алгоритм вычисления $\det A \pmod{p^k}$, использующий $O(n^3 \operatorname{poly}\log m)$ операций.

Задача 4. 3. (4 балла) Пусть A — неотрицательная целочисленная $n \times n$ матрица, элементы которой меньше m. Предложите алгоритм вычисления $\det A \pmod m$, использующий $O(n^3 \operatorname{poly} \log m)$ операций.

Решение, опирающееся на факторизацию m оценивается из 2 баллов.

◇◇

Задача 5. (5×0.5 баллов)

Пусть L язык, состоящий из (кодировок) пар логических ДНФ-формул $\{F_1(\cdot), F_2(\cdot)\}$ от одинакового числа переменных, таких что $F_1 \neq F_2$. Выберите все нужные варианты ответов на следующие пять вопросов, обоснуйте их и обведите соответствующие поля.

При ответе можно пользоваться стандартными гипотезами:

$$\mathbf{P} \subsetneq \mathbf{NP} \cap \textbf{\textit{co}}\text{–}\mathbf{NP};\ \mathbf{NP} \not\subset \textbf{\textit{co}}\text{–}\mathbf{NP};\ \textbf{\textit{co}}\text{–}\mathbf{NP} \not\subset \mathbf{NP}$$

1. $L \in \mathbf{P}$? Да Нет

2. $L \in \mathbf{NP}$? Да Нет

3. $L \in \mathbf{NPC}$? Да Нет

4. $L \in \mathbf{co}\text{–}\mathbf{NP}$? Да Нет

5. $L \in \mathbf{co}\text{–}\mathbf{NPC}$? Да Нет

Критерии.

1. Те же, что и в первом варианте.

Задача 6. (2 балла) Да Нет

Назовем ребро потоковой сети **блокирующим**, если увеличение его пропускной способности приводит к увеличению максимального потока. Покажите, что если ребро (u, v) – блокирующее, то любой минимальный разрез содержит его.

Решение (А.К.): В сети с максимальным потоком в любой минимальный разрез будут входить только насыщенные рёбра. Пусть (u, v) не входит в некоторый минимальный разрез. После его увеличения на единицу по теореме Форда-Фалкерсона в остаточной сети появится увеличивающий путь. Значит, до этого были пути из ненасыщенных рёбер $s \to u$ и $v \to t$. Но разрез должен содержать ребро из любого пути, значит, (u, v) обязан принадлежать разрезу, так как все другие рёбра из пути $s \to u \to v \to t$ ему принадлежать не могли. Противоречие.

Критерии.

1.

Задача 7. (2 балла) Да Нет

Пусть $G(V, E)$ — простой неориентированный граф, множество вершин которого допускает дизъюнктное разбиение на непересекающиеся подмножества $V = S \sqcup T$, такие, что индуцированные подграфы G_S и G_T являются независимыми множествами.

Верно ли, что соответствующий язык всех графов, обладающих таким свойством, принадлежит **NPC**?

По определению, **индуцированный подграф** G_{V_1}, $V_1 \subseteq V(G)$ *имеет вершинами множество* V_1, *а ребрами — все ребра* G *с вершинами из* V_1.

Критерии.

1. Те же, что и в первом варианте.

Задача 8. (2+3 балла)

Задача 8. 1. (2 балла) Пусть G — простой неориентированный граф, в котором степень каждой вершины чётная, и W — натуральное число. Постройте линейный по входу алгоритм проверки, что в G есть цикл, состоящий из $\geq W$ *различных* ребер.
Алгоритм должен быть достаточно подробно описан. Полным баллом оценивается **только линейный алгоритм**

Задача 8. 2. (3 балла) Да Нет
Пусть G — ориентированный граф без петель и кратных дуг и W — натуральное число. Верно ли, что задача проверки, что в G есть цикл, состоящий из $\geq W$ различных дуг, является NP-полной?

Решение (Сергей): Первая задача из **P**. Нужно понять, что это задача про эйлеровость (1 балл) и найти наибольшую по числу ребер компоненту (1 балл). Решение есть, если $W \leq$ числа дуг в такой компоненте, и нет, в противном случае. Для второго пункта годится решение Саши.

Критерии.

1. Из одного балла оценивается понимание того, что это задача об эйлеровых циклах. 1 балл стоит алгоритм, скажем, использующий поиск в глубину, который за линейное время находит разбиение на компоненты связности. Решения нет тогда и только тогда, когда W больше максимального числа ребер в отдельных компонентах.

Решение (А.К.): Да. Сведём гамильтонов цикл в орграфе к данной задаче. Рассмотрим граф G', в котором каждому узлу G сопоставлено два узла v_{in} и v_{out}. При этом есть дуга (v_{in}, v_{out}) и для любой дуги (u, v) в G есть дуга (u_{out}, v_{in}) в G'. Тогда любому циклу без повторов узлов $v_1 \to v_2 \to \cdots \to v_k \to v_1$ в G можно сопоставить цикл без повторов рёбер $v_{1in} \to v_{1out} \to v_{2in} \to v_{2out} \to \cdots \to v_{kin} \to v_{kout} \to v_{1in}$ в G'. В частности, гамильтонову циклу в G будет соответствовать цикл без повторов рёбер в G' длины $2n$.

Но это верно и в обратную сторону, если в G' есть цикл длины $2n$, то в G есть гамильтонов цикл, что формирует наше сведение. Действительно, каждое второе ребро в цикле в G' обязано быть вида $v_{in} \to v_{out}$, а рёбер такого вида всего n – по одному на каждую вершину, значит, если мы нашли цикл длины $2n$, то в нём задействованы все такие дуги и по нему можно восстановить гамильтонов цикл в G.

Критерии.

1.

Задача 9. (3 балла) Да Нет

Пусть G — неориентированный граф с выделенными вершинами s и t, такой, что существует $S \geq 10$ такое, что в G нет пути от s до t длины S.

Длина пути равна числу ребер в нем, а в пути допускается повторение вершин и повторение ребер, т. е. можно, например, возвращаться по ребру, по которому только что был сделан переход.

Верно ли, что соответствующий язык $L = \{(\langle G \rangle, s, t)\}$ принадлежит **NP**?

Критерии.

1. Те же, что и в первом варианте.

Задача 10. (3 балла) Дан неориентированный граф без петель и кратных ребер G, имеющий m рёбер, которым приписаны положительные веса. Раскрасим вершины в два цвета, **легкостью раскраски** назовем наименьший вес ребра между вершинами одного и того же цвета, а если таких рёбер нет, то легкость раскраски считаем равной $+\infty$.

Полным баллом оценивается $O(m \log m)$–алгоритм, находящий раскраску с **наибольшей легкостью**.

Критерии.

1. Те же, что и в первом варианте.

◇◇

Задача 11. (2+2 балла) Рассмотрим циркулянтную матрицу порядка $n+1$, первый столбец которой равен $(c_0, c_1, \ldots, c_n)^T$, т. е. матрицу вида

$$\begin{bmatrix} c_0 & c_n & c_{n-1} & \ldots & c_1 \\ c_1 & c_0 & c_n & \ldots & c_2 \\ c_2 & c_1 & c_0 & \ldots & c_3 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ c_n & c_{n-1} & c_{n-2} & \ldots & c_0 \end{bmatrix}$$

Задача 11. 1. (2 балла) Докажите, что все её собственные значения, домноженные на $\frac{1}{\sqrt{n+1}}$, могут быть найдены умножением матрицы Фурье $F_n = \frac{1}{\sqrt{n+1}} \left(\omega_n^{ij}\right)_{i,j=0}^{n}$ размеров $(n+1) \times (n+1)$, где $\omega_n = e^{\frac{2\pi i}{n}}$ — корень из единицы, на вектор $(c_0, c_n, c_{n-1}, \ldots, c_1)^T$.
Указание. Можно без доказательства пользоваться тем фактом, что любая циркулянтная матрица в базисе из столбцов F имеет диагональный вид.

Решение (П. О.): Требуемое сразу следует из того факта, что F диагонализует любой циркулянт: матрица C в базисе из столбцов F представляет собой диагональную матрицу $\Lambda = diag(\lambda_1, \ldots, \lambda_n) = F^{-1}CF$, откуда $C = F\Lambda F^{-1}$, поэтому $CF = F\Lambda$. Далее транспонируем полученное выражение и учтём симметричность (поэлементную) матрицы Фурье: $FC^T = \Lambda F$. Из этого матричного равенства достаточно взять первый столбец левой и правой части.

Задача 11. 2. (2 балла) Найдите с помощью алгоритма БПФ собственные значения циркулянтной матрицы, первый столбец которой имеет вид $(1, 3, 6, 9)^T$.

Решение (П. О.): Используя формулы из предыдущего пункта, находим спектр C. Для матрицы, порождённой $(1, 3, 6, 9)$ они равны $(19, -5 + 6i, -5, -5 - 6i)$.

Критерии.

1. Критерии, те же, что и в первом варианте.