Программа курса

0) К началу курса необходимо иметь понятие о моделях вычислений в принципе. Необходимы знания про машины Тьюринга (ТФСиА), про алгоритмически разрешимые и неразрешимые задачи (ТФСиА), про детерминированность и недетерминированность (ТРЯП), про простые модели вычислений, которые не способны решать некоторые имеющие решения задачи (ТРЯП). Кроме того, необходимо владеть базовыми понятиями об алгоритмах вообще (ОА), об асимптотическом времени их работы (ОА), об алгоритмах обхода графов (ОА) и также иметь общее представление о модульной арифметике, теореме Эйлера и тому подобном (ОВАиТК).

1) Детерминированная и недетерминированная машина Тьюринга (ДМТ и НМТ). Понятие о вычислении на ДМТ и НМТ, понятие о распознавании языка на ДМТ и НМТ. Временная и пространственная сложность вычислений. Теорема об ускорении. Теоремы об иерархии. Универсальная МТ.

2) Классы DTIME, NTIME, их связь. Классы $\mathcal{P}$ и $\mathcal{NP}$, два определения класса $\mathcal{NP}$, их эквивалентность. $\mathcal{NP}$-трудные и $\mathcal{NP}$-полные языки, связанные с этим задачи. Предположительная трудность $\mathcal{NP}$-трудных задач.

3) Сводимость по Тьюрингу, по Куку и по Карпу. Использование сводимости по Карпу для доказательства $\mathcal{NP}$-полноты. Теорема Кука-Левина.

4) Со-классы, классы со$\mathcal{P}$, со$\mathcal{NP}$, их связь с $\mathcal{P}$ и $\mathcal{NP}$, примеры задач из класса со$\mathcal{NP}$. Полиномиальная иерархия.

5) Самосводимость и задачи поиска. Аппроксимационные алгоритмы для решения $\mathcal{NP}$-полных задач. Вычисления с оракулом.

6) Классы DSPACE, NSPACE, их связь с временными классами. Класс $\mathcal{PSPACE}$. Теорема Савича.

7) Класс $\mathcal{L}$, примеры задач из этого класса. Класс $\mathcal{NL}$, логарифмическая сводимость. $\mathcal{NL}$-полнота. Теорема Иммермана-Селепченьи.

8) Неравномерная и схемная сложность. Класс сложности $\mathcal{P}$/poly. Распознавание языка семейством схем, класс SIZE.

9) Вероятностная машина Тьюринга, понятие о вероятностном вычислении. Классы $\mathcal{RP}$ и со$\mathcal{RP}$, их связь с $\mathcal{NP}$ и со$\mathcal{NP}$. Классы $\mathcal{BPP}$ и $\mathcal{PP}$, их связь и связь с невероятностными классами.

10) Амплификация и независимость вероятностных классов от значений вероятностей. Дерандомизация вероятностных алгоритмов.

11) Интерактивные протоколы и интерактивные доказательства. Публичные и приватные протоколы. Доказательство с нулевым разглашением.

Литература

1. “Computational Complexity: A Modern Approach” by Sanjeev Arora and Boaz Barak
2. “Computational Complexity” by Christos Papadimitriou
3. “Computational Complexity: A Conceptual Perspective” by Oded Goldreich
4. “Complexity of Algorithms” by László Lovász
5. “Introduction to the Theory of Computation” by Michael Sipser
6. “The Nature of Computation” by Cristopher Moore and Stephan Mertens

7. “Mathematics and Computation” by Avi Wigderson
8. “Computers and Intractability: A Guide to the Theory of NP-Completeness” by Michael Garey and David Johnson
9. “Formal languages and their relation to automata” by John Hopcroft and Jeffrey Ullman
10. “Сложность вычислений” Мусатов Д.В., черновик

Задачи

1. Дайте максимально точные асимптотические оценки сверху и снизу для $T(n)$ (считать $T(n)$ константой для $n \leqslant 3$).

а) $T(n) = T(\frac{n}{2}) + T(\frac{n}{3}) + T(\frac{n}{6}) + cn$,

б) $T(n) = T(\frac{n}{2}) + T(\frac{n}{4}) + T(\frac{n}{6}) + cn$,

в) $T(n) = T(\sqrt{n}) + 1$.

2.

1) Покажите, что язык PAL = $\{w \mid w \in \{0,1\}^*, w$ – палиндром$\}$ распознается за $T(n) = O(n)$ на многоленточной МТ.

2) Докажите, что любая одноленточная МТ, распознающая PAL, работает за время $T(n) = \Omega(n^2)$

3. Пусть M работает за время $T(n)$ и существует такое n_0, что $T(n_0) < n_0 + 1$. Докажите, что:

1) M не читает символ номер $n_0 + 1$ ни у какого входа;

2) M работает за время $O(1)$;

3) M распознаёт регулярный язык.

4. Постройте одноленточную МТ, которая переводит вход 1^n в выход 1^{2n}, т.е. удваивает слово, заданное в унарном алфавите. Время работы $O(n \log n)$.

5. Чтобы не рассматривать различные входные алфавиты и алфавиты МТ, а также чтобы уметь задавать вопросы о произвольных конструктивно заданных объектах, а не только строках, достаточно научиться кодировать все данные в бинарном алфавите.

Предъявите какую-нибудь эффективную (вычислимую и при этом желательно быструю) процедуру получения строки над алфавитом $\{0,1\}$ по объекту и, наоборот, получения объекта по строке для следующих множеств объектов. Заметьте, что декодирование (получение объекта по строке) может быть сопряжено с технической трудностью: не все строки при кодировании становятся образом какого-то объекта – среди строк появляется “мусор”.

Объекты:

а) пары бинарных строк с разделителями;

б) натуральные числа;

в) булевы формулы;

г) рациональные числа (можно ли вещественные?);

д) матрицы;

е) графы;

ё) машины Тьюринга.

6. Рассмотрим модификацию МТ (здесь подразумеваются одноленточные односторонние МТ) – прыгающую МТ. Прыгающая МТ аналогична обычной МТ, во всём, кроме возможных действий: на каждом шаге прыгающая МТ может только сдвинуть головку на одну клетку вправо (как обычная МТ) или сдвинуть головку на символ номер нуль (перепрыгнуть до упора влево). Докажите, что для любой МТ существует эквивалентная ей прыгающая МТ. Оцените время её работы.

7. Доказать, что класс $\mathcal{P}$ замкнут относительно операций:

– объединения,

– пересечения,

– конкатенации,

– итерации (звёздочки Клини),

– дополнения,

– чётной итерации.

8. Доказать что языки принадлежат $\mathcal{P}$:

– язык всех несвязных графов без циклов (кодировка графа на ваше усмотрение),

– язык всех квадратных матриц на целых числах, в которых есть подматрица размера 2023×2023, заполненная нулями,

– язык всех графов, в которых существует эйлеров цикл и как минимум пять вершин имеют нечётную степень.

9.

1) доказать эквивалентность двух определений класса $\mathcal{NP}$,

2) доказать, что $\mathcal{P} \subset \mathcal{NP} \cap \mathrm{co}\mathcal{NP}$,

3) доказать, что $\mathcal{P} = \mathcal{NP}$ влечёт $\mathcal{NP} = \mathrm{co}\mathcal{NP}$,

4) доказать, что $\mathrm{co}\mathcal{NP} \neq \left(2^{\Sigma^*} \setminus \mathcal{NP}\right)$,

5) доказать, что $\mathcal{NP} \subset \bigcup_{k=0}^{\infty} \mathrm{DTIME}\left(2^{n^k}\right)$.

10. Пусть $L \in \mathrm{co}\mathcal{NP}$, тогда $\overline{L} \in \mathcal{NP}$ по определению. Тогда для $\overline{L}$ есть МТ M такая, что $M(x,u) = 1$ для $x \in \overline{L}$ и $M(x,u) = 0$ для $x \notin \overline{L}$. Но тогда можно определить новую МТ M', поменяв состояния M на противоположные. Тогда $M'(x,u) = 1$ для $x \notin \overline{L}$, т.е. для $x \in L$ и наоборот. Тогда $L \in \mathcal{NP}$ и $\mathcal{NP} = \mathrm{co}\mathcal{NP}$.

Где ошибка в рассуждениях?

11. Почему в определении $\mathrm{co}\mathcal{NP}$ мы не меняем существование МТ на всеобщность (для любой МТ), существование полинома на время работы МТ на всеобщность (для любого полинома) и существование полинома на ограничение длины сертификата на всеобщность (для любого полинома) по сравнению с определением $\mathcal{NP}$?

12. Всюду под «можно ли» подразумевается следующее: останется ли при такой замене класс языков $\mathcal{NP}$ неизменным.

1) В определении стоит $y \in \Sigma^{q(|x|)}$. Можно ли смягчить требование, положив $y \in \Sigma^*, |y| \leqslant q(|x|)$?

2) В определении МТ останавливается за $p(|x|+|y|)$ шагов. Можно ли убрать из требования y, оставив $p(|x|)$ шагов?

3) Можно ли после предыдущих упрощений вообще убрать полиномиальное ограничение на y, получив такое определение

$L \in \mathcal{NP}$, если: существует ДМТ M, существует полином $p(n)$, такие, что

для любых $x \in \Sigma^*$, $y \in \Sigma^*$ машина M останавливается на входе (x, y) не более, чем за $p(|x|)$ шагов и

$$(x \in L) \Leftrightarrow \exists y\,(y \in \Sigma^* \wedge M(x, y) = 1)$$

13.

1) Привести пример языка L такого, что он не принадлежит $\mathcal{P}$.

2) Привести пример языка L такого, что он не принадлежит $\mathcal{P}$, а L^* принадлежит $\mathcal{P}$.

14. Доказать что языки принадлежат $\mathcal{NP}$.

1) Язык пар графов (G, H) таких, что графы G и H изоморфны.

2) Язык всех графов, содержащих эйлеров путь.

3) Язык всех графов, содержащих гамильтонов путь.

4) Язык всех взвешенных орграфов, в которых нет цикла отрицательной длины.

5) Язык всех составных чисел (записаны в десятичной системе счисления).

6) CNF–SAT

15. Для принадлежности классу $\mathcal{NP}$ не требуется никакого алгоритма получения сертификата, лишь его существование. Почему в таком случае каждый разрешимый язык L не принадлежит $\mathcal{NP}$, ведь для любого слова x можно предъявить очень простой сертификат: 1, если $x \in L$, или 0, если $x \notin L$?

16. Проблема останова МТ неразрешима — это известно (по крайней мере, из курса ТФСиА). Однако, для слов, принадлежащих соответствующему языку, существует простейший сертификат — для пары (МТ, слово) сертификатом будет число N — число шагов до останова. Для любой пары (МТ, слово), такой, что МТ останавливается на этом слове, сертификат очевидно существует. Для любой пары, не принадлежащей языку, никакой сертификат не подойдёт. Следовательно, проблема останова принадлежит классу $\mathcal{NP}$, а значит разрешима на НМТ, т.е. разрешима. Где ошибка в рассуждениях?

17. Докажите, что классы $\mathcal{NP}$ и co$\mathcal{NP}$ замкнуты относительно операций:

а) объединения,

б) пересечения,

в) конкатенации,

г) итерации (звёздочки Клини).

18. Язык PRIMES — это язык простых чисел, двоичная запись натурального числа x принадлежит языку PRIMES тогда и только тогда, когда x простое число. Доказать, что

– PRIMES $\in$ co$\mathcal{NP}$,

– PRIMES $\in \mathcal{NP}$.

19. Докажите, что язык

FACTOR = $\{(n, k) \mid n$ содержит делитель, больший 1, но не превосходящий $k\}$

принадлежит классу $\mathcal{NP} \cap \text{co}\mathcal{NP}$.

20. Докажите, что языки принадлежат $\text{co}\mathcal{NP}$

1) TAUT – язык, состоящий из описаний булевых тавтологий,

2) Язык, состоящий из пар (G, k) где G – описание графа, такого, что для любых k вершин найдется ребро, соединяющее хотя бы 2 из них.

3) Язык описаний графов, в которых есть клика на 2023 вершинах (клика – это подмножество вершин графа, таких, что каждая соединена ребром с каждой).

4) PLANARITY – язык описаний планарных графов.

21. Пусть ExactClique = $\{(G, k) \mid G$ содержит клику размера k и не содержит клику размера $k + 1\}$.

1) Докажите, что ExactClique $\in \Sigma_2^p$.

2) Докажите, что ExactClique $\in \Pi_2^p$.

22. Докажите, что

1) для любого i выполняется $\text{co}\,\Sigma_i^p = \Pi_i^p$;

2) для любого i выполняется $\Sigma_i^p \subset \Sigma_{i+1}^p$, $\Sigma_i^p \subset \Pi_{i+1}^p$, $\Pi_i^p \subset \Sigma_{i+1}^p$, $\Pi_i^p \subset \Pi_{i+1}^p$;

3) если для некоторого $i \geqslant 1$ выполняется $\Sigma_i^p = \Pi_i^p$, то $\mathcal{PH} = \Sigma_i^p$ (тогда говорят, что полиномиальная иерархия схлопывается или коллапсирует на уровень i).

23. Докажите, что язык Σ_i^p – SAT является полным в Σ_i^p. Язык Σ_i^p – SAT определяется как набор истинных формул вида $\exists x_1 \forall x_2 \ldots Q_i x_i\, \varphi(x_1, x_2, \ldots, x_i)$, здесь φ есть КНФ. Здесь Q_i есть $\forall$, если i чётно, и $\exists$, если i нечётно.

24. 1) Верно ли, что если $L \in \mathcal{NPC}$ и $L \in \text{co}\mathcal{NP}$, то $\mathcal{NP} = \text{co}\mathcal{NP}$?

2) Верно ли, что если $L \in \mathcal{NP}$ и $L \in \text{co}\mathcal{NPC}$, то $\mathcal{NP} = \text{co}\mathcal{NP}$?

3) Верно ли, что язык $L \in \mathcal{NPC}$ тогда и только тогда, когда $\overline{L} \in \text{co}\mathcal{NPC}$?

4) Доказать, что если $\mathcal{P} = \mathcal{NP}$, то любой нетривиальный язык $\mathcal{NP}$-трудный. (Что насчет тривиальных языков?)

5) Верно ли, что если $L_1 \leqslant_p L_2$, то $\overline{L}_1 \leqslant_p \overline{L}_2$?

25. По аналогии с обычной SAT определим задачу DNF–SAT: дана ДНФ, нужно проверить, выполнима ли она (т.е. дан язык всех выполнимых ДНФ, нужно построить распознаватель для него)

1) Придумать полиномиальный алгоритм решения DNF–SAT (это очень просто).

2) Построим сводимость CNF–SAT к DNF–SAT: в формуле КНФ раскрываем скобки в силу дистрибутивности операций конъюнкции и дизъюнкции. Доказать корректность сводимости.

3) Поскольку мы свели $\mathcal{NP}$–полную задачу к полиномиально разрешимой, то $\mathcal{P} = \mathcal{NP}$. Есть ли ошибка в рассуждениях?

26. Построить полиномиальную сводимость:

а) Hampath к Hamcycle

б) Hamcycle к Hampath

27. Мы рассматривали неориентированные графы на гамильтоновость. Рассмотрим теперь гамильтоновы ориентированные графы. Всюду приставка Dir означает ориентированность графа.

Построить полиномиальную сводимость:

а) Hampath к DirHampath

б) DirHampath к Hampath

Можете повторить то же самое для циклов.

28. Для каждого натурального k определим язык k–Clique всех графов, в которых есть клика размера k.

1) Язык 1–Clique полиномиален?

2) Язык 2–Clique полиномиален?

3) Язык 3–Clique полиномиален?

4?) Для каких k язык k–Clique $\mathcal{NP}$-полон?

5??) Известно, что Clique $\in \mathcal{NPC}$, но из входа (G, k) можно сперва выделить k, а затем запустить алгоритм решения соответствующего k–Clique. Как тогда может быть, что Clique $\in \mathcal{NPC}$?

6!) Доказать, что язык графов, имеющих клику ровно на половине вершин, $\mathcal{NP}$-полон.

29. Доказать, что следующие языки принадлежат $\mathcal{NPC}$.

1) Графы, имеющие вершинное покрытие ровно на половине вершин.

2) Пары граф G и число k такие, что существует простой путь в графе G длины как минимум k.

3) Графы G, в которых есть простой путь, проходящий ровно по разу через все, кроме 2023 вершин.

4) Графы G, в которых есть минимальное остовное дерево со степенью всех вершин не более 42.

30. Описать полиномиальные алгоритмы для решения задач:

– 2–Partition

– 2–Color

– 2–SAT

31. Докажите, что язык Taut всех булевых тавтологий со$\mathcal{NP}$-полон. (Можно ограничиться формулами в ДНФ.)

32. Напомним, задача останова Halt неразрешима.

1) Верно ли, что Halt $\in \mathcal{NPC}$?

2) Верно ли, что Halt $\in \mathcal{NP}$-hard?

33. Замкнут ли $\mathcal{NPC}$ относительно:

а) объединения,

б) пересечения,

в) конкатенации,

г) итерации (звёздочки Клини)?

д) Если $L_1 \in \mathcal{NPC}$ и $L_2 \in \mathcal{NPC}$, то верно ли, что $L_1 \times L_2 \in \mathcal{NPC}$?

34. Большому количеству $\mathcal{NPC}$ задач и, соответственно, $\mathcal{NPC}$ языков соответствуют задачи поиска. Приведём пример:

Clique = $\{(G,k) \mid G$ содержит клику размера $k\}$ – задача распознавания и одновременно язык. Дана пара (G,k), выясним «да, принадлежит» или «нет, не принадлежит».

ExactClique = $\{(G,k) \mid G$ содержит клику размера k и не содержит клику размера $k+1\}$ – тоже задача распознавания и одновременно язык. Дана пара (G,k), выясним «да, принадлежит» или «нет, не принадлежит».

MaxSizeOfClique$(G) = k$ – задача поиска, на вход подаётся граф G, на выходе нужно получить размер макисмальной клики в графе G – число k.

MaxClique$(G) = G'$ – задача поиска, на вход подаётся граф G, на выходе нужно получить клику максимального размера, являющуюся подграфом G – назовём её G'.

Заметим, Clique – это язык (либо предикат), MaxClique – это не язык, это функция, которая может быть вычислена некоторым алгоритмом.

Рассмотрим задачи Clique, ExactClique, MaxSizeOfClique, MaxClique. Докажите, что если для одной из них есть полиномиальный алгоритм, то он есть для каждой из этих задач. Не забывайте, что последние две не являются задачами распознавания – полиномиальность алгоритма для них определяется естественным способом.

35. Покажите, что если HamPath-s-t $\in \mathcal{P}$, то за полиномиальное время можно не только определить, что в графе существует гамильтонов путь из s в t, но и найти его.

Повторите то же самое для HamPath и HamCycle.

36. Покажите, что если 3–Color $\in \mathcal{P}$, то за полиномиальное время можно не только определить, что граф допускает раскраску вершин в три цвета, но и найти какую-то 3-раскраску, если она существует.

37. Докажите, что для оптимизационной версии TSP аппроксимационная отсечка $\varepsilon = 1$, т.е. полиномиальная аппроксимация невозможна ни с какой точностью ε.

38. Докажите, что для оптимизационной версии метрического TSP существует полиномиальная аппроксимация с точностью $\varepsilon = \frac{1}{3}$ (начните с $\varepsilon = \frac{1}{2}$).

39. Докажите, что для оптимизационной версии VertexCover существует полиномиальная аппроксимация с точностью $\varepsilon = \frac{1}{2}$.

40. Докажите, что для оптимизационной версии SubsetSum аппроксимационная отсечка $\varepsilon = 0$. Постройте соответствующую FPTAS.

41. Докажите

1) Каждый $\mathcal{PSPACE}$-трудный язык также $\mathcal{NP}$-трудный.

2) Если существует $\mathcal{PSPACE}$-полный язык, принадлежащий $\mathcal{NP}$, то $\mathcal{NP} = \mathcal{PSPACE}$.

3) Если каждый $\mathcal{NP}$-трудный язык также $\mathcal{PSPACE}$-трудный, то $\mathcal{NP} = \mathcal{PSPACE}$.

42. Игра в города для двух игроков выглядит следующим образом:

– выбирается название начального города: “Долгопрудный”,

– первый игрок называет город, название которого начинается с последней буквы предыдущего города: “Йошкар-Ола”,

– второй игрок называет город, название которого начинается с последней буквы предыдущего города: “Абакан”,

– первый игрок снова делает ход и так далее...

Запрещено использовать город повторно. Проигрывает тот игрок, который не может сделать ход.

Определим обобщённую игру в города. Дан конечный язык L над фиксированным конечным алфавитом и начальный “город” $c \in L$.

Язык Geo – язык всех пар (L, c), что в игре в города со списком городов L и начальным городом c при правильной игре побеждает игрок номер 1 (т.е. тот игрок, который первым называет город после города c).

Докажите, что Geo $\mathcal{PSPACE}$-полный язык.

43. Рассмотрим игру Hex на поле $n \times n$. Правила таковы (hooray to Wiki):

> Hex is played on a rhombic grid of hexagons. Each player has an allocated color, conventionally Red and Blue. Each player is also assigned two opposite board edges. The hexagons on each of the four corners belong to both adjacent board edges. The players take turns placing a stone of their color on a single cell on the board. Red goes first. Once placed, stones are not moved, replaced, or removed from the board. Each player's goal is to form a connected path of their own stones linking their two board edges. The player who complete such a connection wins the game.

Определим язык Hex – язык всех пар (B, p), что в игре в Hex с начальной конфигурацией доски B побеждает игрок p. Конфигурация доски – матрица размера $n \times n$, заполненная нулями, единицами и двойками в зависимости от того, какой камень лежит в соответствующем поле доски (и лежит ли).

Докажите, что Hex $\in \mathcal{PSPACE}$. Оцените сложность по памяти для алгоритма распознавания.

44.

1) Докажите, что язык ALL-NFA = $\{\langle M\rangle |\ M$ – НКА и $L(M) = \Sigma^*\}$ является $\mathcal{PSPACE}$-полным.

2) Что можно сказать о языке ALL-DFA = $\{\langle M\rangle |\ M$ – ДКА и $L(M) = \Sigma^*\}$?

45. Доказать, что языки лежат в $\mathcal{L}$

– ADD = $\{a\#b\#c \mid c = a + b\}$,

– MULT = $\{a\#b\#c \mid c = a \cdot b\}$,

– язык всех неориентированных графов, содержащих цикл,

– язык всех неориентированных графов, не содержащих цикл.

46. Доказать, что в $\mathcal{L}$ лежат

– язык правильных скобочных выражений,

– язык правильных скобочных выражений с двумя типами скобок,

– язык всех палиндромов.

47. 1) Доказать, что два определения класса $\mathcal{NL}$ эквивалентны.

2) Какой класс получится, если заменить полиномиальную длину сертификата на логарифмическую и убрать ограничение на чтение сертификата единожды?

3) Какой класс получится, если заменить полиномиальную длину сертификата на логарифмическую, но оставить ограничение на чтение сертификата единожды?

4) Какой класс получится, если оставить полиномиальную длину сертификата и убрать ограничение на чтение сертификата единожды?

48. Пусть $L_1 \leqslant_{\log} L_2$, $L_2 \leqslant_{\log} L_3$

1) доказать, что если $L_2 \in \mathcal{L}$, то $L_1 \in \mathcal{L}$,

2) доказать, что если $L_2 \in \mathcal{NL}$, то $L_1 \in \mathcal{NL}$,

3) доказать, что $L_1 \leqslant_{\log} L_3$.

49. Доказать, что stCON лежит в $\mathcal{NL}$

50. Доказать, что для любой "хорошей" f выполняется

$$\mathrm{DTIME}(f) \subset \mathrm{NTIME}(f) \subset \mathrm{DSPACE}(f) \subset \mathrm{NSPACE}(f) \subset \mathrm{DTIME}\left(2^{O(f)}\right)$$

51. Известно, что USTCON $\in \mathcal{L}$. Для каждого из языков доказать, что он лежит в $\mathcal{L}$ и что он эквивалентен по сложности USTCON (язык сводится к USTCON и обратно USTCON сводится к языку)

– язык всех не двудольных графов,

– язык всех двудольных графов,

– язык пар граф G и ребро графа e, таких, что в G есть цикл, проходящий через e

52. Доказать принадлежность $\mathcal{NL}$

– языка двудольных графов,

– языка 2–Color.

53. Доказать $\mathcal{NL}$-полноту языков

– ShortestPath = $\{(G, k, s, t) |$ G – орграф, кратчайший путь из s в t в графе G имеет длину ровно $k\}$

здесь лучше доказать двумя способами – с помощью свойств языков $\mathcal{NL}$ и co$\mathcal{NL}$ и построением явной НМТ (или ДМТ с сертификатом), распознающей язык,

– всех сильносвязных орграфов,

– всех орграфов, содержащих цикл,

– всех орграфов, не содержащих цикла (DAG),

– 2-SAT,

– всех пар (A, ω), где НКА A принимает слово ω.

54. Доказать, что

1) $\mathcal{P} \subset \mathcal{P}/\mathrm{poly}$,

2) $\mathcal{P} \neq \mathcal{P}/\mathrm{poly}$ и даже $\mathcal{P} \neq \mathcal{P}/1$,

3) $\mathcal{P}/\mathrm{poly}$ содержит разрешимый язык, не принадлежащий $\mathcal{P}$.

55. Доказать, что если в определении $\mathcal{P}/\mathrm{poly}$ ограничиться полиномиально вычислимыми α (т.е. такими, что существует полиномиальная ДМТ на входе n выдающая α_n), такое ограничение даст класс $\mathcal{P}$.

56. 1) Язык называется унарным, если все его слова имеют вид 1^k. Доказать, что любой унарный язык принадлежит $\mathcal{P}/\mathrm{poly}$

2) Пусть язык L обладает следующим свойством: любые два слова x и y одинаковой длины либо оба принадлежат L, либо оба не принадлежат L. Доказать, что $L \in \mathcal{P}/\mathrm{poly}$.

57. Пусть язык $A \in \mathcal{P}/\mathrm{poly}$, а для языка B выполняется

$$\forall n \in \mathbb{N} \quad |(A \triangle B) \cap \{0,1\}^n| \leqslant n$$

Докажите, что $B \in \mathcal{P}/\mathrm{poly}$.

58. Рассмотрим следующий алгоритм равномерного перемешивания колоды из n карт (это значит, что вероятность получить любое конкретное перемешивание одинакова):

1) возьмём колоду и пронумеруем карты сверху вниз от 1 до n;

2) далее работаем по шагам, на каждом шаге:

2.1) берём карту с вершины стопки,

2.2) в оставшейся колоде из $n-1$ карты равновероятно выбираем случайное место для карты (n способов – на верх колоды или под какую-то из карт колоды),

2.3) вставляем взятую карту в выбранное случайное место (карт в колоде снова n);

3) алгоритм заканчивает свою работу после того, как цикл сработал на карте номер $n-1$, т.е. карта номер $n-1$ была взята сверху и вставлена в случайное место.

а) Доказать, что представленный алгоритм работает корректно, т.е. перемешивает колоду равномерно.

б) Найти математическое ожидание количества шагов при выполнении алгоритма, т.е. числа вставок карты в колоду.

59. Назовём классом $\mathcal{BPP}_{(\varepsilon_1,\varepsilon_2)}$ класс языков, распознаваемых ПВМТ M, причём

$x \in L \Rightarrow \mathbb{P}(M(x) = 1) \geqslant 1 - \varepsilon_1$,

$x \notin L \Rightarrow \mathbb{P}(M(x) = 0) \geqslant 1 - \varepsilon_2$.

Напомним, что $\mathcal{BPP} = \mathcal{BPP}_{(1/3,1/3)}$ по определению.

Доказать, что для любых $\varepsilon_1, \varepsilon_2 < 1/2$ выполняется $\mathcal{BPP} = \mathcal{BPP}_{(\varepsilon_1,\varepsilon_2)}$

60. Назовём классом $\mathcal{BPP}_p$ класс языков, распознаваемых ПВМТ M, причём существует положительный полином $p(n)$ и

$x \in L \Rightarrow \mathbb{P}(M(x) = 1) \geqslant \frac{1}{2} + \frac{1}{p(|x|)}$,

$x \notin L \Rightarrow \mathbb{P}(M(x) = 0) \geqslant \frac{1}{2} + \frac{1}{p(|x|)}$.

Доказать, что $\mathcal{BPP} = \mathcal{BPP}_p$

61. Доказать, что

a) $\mathcal{RP} \subset \mathcal{NP}$, $\mathrm{co}\,\mathcal{RP} \subset \mathrm{co}\,\mathcal{NP}$

b) $\mathcal{RP} \subset \mathcal{BPP}$, $\mathrm{co}\,\mathcal{RP} \subset \mathcal{BPP}$

c) $\mathcal{BPP} \subset \mathcal{PP}$, $\mathcal{NP} \subset \mathcal{PP}$

62. Классу языков **X** принадлежат все такие языки, что для них существует ПВМТ M с тремя завершающими состояниями («да», «нет», «ответ неясен»), причём верно следующее. На любом входе x ПВМТ M с некоторой вероятностью $p_{\text{хз}} < 1$ завершает работу в состоянии «ответ неясен», с вероятностью $p_e < \frac{1-p_{\text{хз}}}{2}$ ошибается (неправильно определяет принадлежность/непринадлежность x языку) и с вероятностью $1 - p_{\text{хз}} - p_e$ правильно определяет принадлежность/непринадлежность x языку. Значения $p_{\text{хз}}, p_e$ могут зависеть от x.

Класс **X** равен какому-то известному Вам классу языков. Какому и почему?

63. MAXCUT – это задача следующего вида: дан граф G, нужно найти разбиение вершин графа на два множества так, чтобы число рёбер, у которых концы лежат в разных множествах, максимально.

1) Придумать простой вероятностный алгоритм решения MAXCUT.

2) Дерандомизировать алгоритм, используя метод условных матожиданий.

64. MAXSAT – это задача следующего вида: дана булева формула в виде КНФ, нужно найти набор переменных, на котором выполняется наибольшее возможное число дизъюнктов.

1) Придумать простой вероятностный алгоритм решения MAXSAT.

2) Дерандомизировать алгоритм, используя метод условных матожиданий.

65. Пусть $\mathcal{ZPP} = \mathcal{RP} \cap \operatorname{co}\mathcal{RP}$. Доказать, что следующие утверждения эквивалентны:

1) $L \in \mathcal{ZPP}$;

2) существует вероятностная машина Тьюринга, выдающая на слове x правильный ответ с вероятностью единица, при этом не *худшее* время работы у неё полиномиально, а **ожидаемое** время работы полиномиально;

3) существует ПВМТ, которая на слове x с вероятностью $1/2$ отвечает верно (правильно определяется принадлежность или непринадлежность слова языку), а с вероятностью $1/2$ отвечает "ответ неясен".

66. Лемма Шварца-Зиппеля. Пусть $p \in F[x_1, x_2, \ldots, x_n]$ – ненулевой полином от n переменных степени $d \geqslant 0$ над полем F. Пусть S конечное подмножество F и пусть элементы $r_1, r_2, \ldots, r_n$ были выбраны из S равномерно и независимо друг от друга.

Тогда

$$\mathbb{P}[p(r_1, r_2, \ldots, r_n) = 0] \leqslant \frac{d}{|S|}$$

Доказать лемму. Решить с помощью леммы задачу PIT (polynomial identity testing). Дан полином p, верно ли, что после раскрытия всех скобок и приведения его к сумме мономов все коэффициенты перед мономами будут равны нулю? В данном случае "решить" – это "предъявить эффективный вероятностный алгоритм".

67. Пусть мы запускаем $\mathcal{PP}$ или $\mathcal{BPP}$ алгоритм t раз. $X_i = 1$, если ответ алгоритма на запуске i правильный и 0 в противном случае.

Предположим, после t запусков мы принимаем решение о принадлежности слова языку на основании "мнения большинства". На основе оценки Чернова докажите, что класс $\mathcal{BPP}$ "эффективен", а класс $\mathcal{PP}$ "не эффективен", т.е. что для получения малой вероятности ошибки (пусть полиномиально малой или экспоненциально малой) в первом случае требуется полиномиальное число запусков алгоритма, а во втором полиномиального числа запусков может не хватить. (Полиномиальное число – как и время работы алгоритма – считается от длины входа).

68. Определим задачу EZE (evaluate to zero everywhere). Дан полином p от n переменных степени $d \geqslant 0$ над полем F, верно ли, что при любом выборе элементов $r_1, r_2, \ldots, r_n$ из поля F, значение $p(r_1, r_2, \ldots, r_n) = 0$?

Доказать, что EZE $\in \mathrm{co}\mathcal{NP}$-hard.

Ранее было доказано, что PIT эффективно разрешим. Следует ли из вышесказанного, что $\mathrm{co}\mathcal{RP} = \mathrm{co}\mathcal{NP}$?

69. Назовём классом $\mathcal{PP}_{\text{more equal}}$ класс языков, распознаваемых ПВМТ M, причём

$x \in L \Rightarrow \mathbb{P}(M(x) = 1) > 1/2$,

$x \notin L \Rightarrow \mathbb{P}(M(x) = 0) \geqslant 1/2$.

Докажите, что $\mathcal{PP}_{\text{more equal}} = \mathcal{PP}$.

70. Назовём классом $\mathcal{PP}_{\text{even more equal}}$ класс языков, распознаваемых ПВМТ M, причём

$x \in L \Rightarrow \mathbb{P}(M(x) = 1) \geqslant 1/2$,

$x \notin L \Rightarrow \mathbb{P}(M(x) = 0) \geqslant 1/2$.

Докажите, что $\mathcal{PP}_{\text{even more equal}} = 2^{\Sigma^*}$.

71. Задача S состоит в следующем: по данному неориентированному графу G найти такую раскраску вершин графа в четыре цвета, что количество разноцветных рёбер в графе (рёбер, концы которых покрашены в различные цвета) максимально.

1) Придумайте вероятностный алгоритм для приближённого решения задачи S с точностью $\frac{3}{4}$.

2) Дерандомизируйте полученный алгоритм методом условных математических ожиданий.

72. Студент (**С**) и Преподаватель (**П**) играют в следующую игру.

На поле из $n+1$ клетки, нумерованной от 0 до n, могут располагаться k различных фишек.

До начала ходов все k фишек расположены на поле номер 0.

На своём ходу С выбирает два подмножества фишек A и B из находящихся на текущий момент на доске. При этом

– $A \neq \varnothing$, $B \neq \varnothing$,

– $A \cap B = \varnothing$,

– $A \cup B$ не обязано покрывать все фишки на доске.

На своём ходу П выбирает одно из двух подмножеств A или B. При этом

– все фишки из выбранного подмножества удаляются с доски и более в игре не участвуют,

– каждая фишка из невыбранного подмножества перемещается вперёд на одно поле на доске (номер поля, на котором она располагается, увеличивается на 1),

– каждая фишка, не входящая ни в A, ни в B, остаётся на месте.

Начинает игру **С**. **Игра продолжается** до тех пор, пока не выполнится одно из условий:

– хотя бы одна фишка попадает на поле с номером n – в этом случае выигрывает **С**;

– **С** не может сделать ход (на поле не осталось фишек, либо осталась одна фишка не на поле с номером n) – в этом случае выигрывает **П**.

1) Пусть $k \geqslant 2^n$. Доказать, что у **С** существует выигрышная стратегия.

2) Пусть $k < 2^n$. С помощью вероятностного алгоритма доказать, что у **П** существует выигрышная стратегия.

3) Дерандомизировать вероятностный алгоритм, получив при $k < 2^n$ детерминированную выигрышную стратегию для **П**.

73. Вам дана нечестная монета – вы не знаете, насколько она нечестна, т.е. вы не знаете вероятности выпадения орла и решки при броске, вы знаете лишь, что обе эти вероятности отличны от нуля.

1) Придумать алгоритм, симулирующий бросок честной монеты – равновероятное получение единицы или нуля. В качестве единственного источника случайных значений вы можете использовать нечестную монету.

2) Вычислить математическое ожидание числа «бросков» нечестной монеты для вашего алгоритма.

Формально, вы моделируете бернуллиевскую случайную величину с параметром $1/2$ при помощи бернуллиевской случайной величины с неизвестным параметром $p \in (0, 1)$.

74. Вам дана честная монета – вероятности выпадения орла и решки при броске одинаковы.

1) Придумать алгоритм, симулирующий бросок нечестной монеты с заданной вероятностью выпадения орла и решки. В качестве единственного источника случайных значений вы можете использовать честную монету.

2) Вычислить математическое ожидание числа «бросков» честной монеты для вашего алгоритма.

Формально, вы моделируете произвольную бернуллиевскую случайную величину с параметром $p \in (0, 1)$ при помощи бернуллиевской случайной величины с параметром $1/2$.