Московский физико-технический институт (НИУ)
Физтех-школа прикладной математики и информатики
Сложность вычислений, осень 2023
Программа курса, версия от 13 сентября 2023 г.

1 Основные ссылки по курсу

Все материалы по курсу будут появляться в Телеграм-канале

`https://t.me/diht_complexity`

и сохраняться в общей папке

(ссылка на дропбокс),

также их можно обсудить в чате

(ссылка-приглашение в чат).

В течение сентября будет создана табличка с оценками за контрольные работы и домашние задания.

Планируется, что будут выкладываться следующие материалы:

- Черновик книги, в основном покрывающей конспекты лекций (по этому и ряду других курсов). Планируется добавлять отдельные фрагменты в течение семестра;
- Листочки с задачами для семинарских занятий;
- Частичные решения этих листочков;
- Тренировочные варианты контрольных;
- Выданные варианты контрольных;
- Индивидуально подготовленные домашние задания;

2 Зачем нужен этот курс

В курсе теории алгоритмов мы познакомились с различием разрешимых и неразрешимых задач. Но для практики мало, чтобы задача была разрешимой в принципе. Требуется, чтобы она была разрешима за приемлемое время. Для этого нужно ввести понятие *эффективной* вычислимости и (в идеале) научиться отделять эффективно решаемые задачи от тех, для которых такого решения нет.

Определять эффективность вычислений можно самыми разными способами. Как правило, сначала вводится вычислительная модель (например, детерминированные или вероятностные вычисления), потом говорится, какие ресурсы мы будем измерять в этой модели (например, время работы, использованную память или потребное число случайных битов), затем вводится некоторое ограничение на эти ресурсы (например, время работы должно быть полиномиальным от длины входа) и рассматривается класс всех задач, решаемых в этой модели с таким ограничением (в изложенном примере это будет

класс **P**). Возникает целый «зоопарк» сложностных классов, которые можно условно поделить на доступные для эффективных вычислений и недоступные. Основной теоретический вопрос — как связаны одни классы с другими и где точно проходит граница между эффективными и неэффективными вычислениями.

В ходе курса мы изучим фрагмент этого «зоопарка» примерно из 30 классов. Мы познакомимся с базовыми моделями вычислений, видами ресурсов, методами их подсчёта и основными классами. Мы научимся оценивать сложность задач через классификацию их по этим классам, докажем основные теоремы о соотношении классов и очень быстро упрёмся в открытые проблемы, которых в этой области всё ещё гораздо больше, чем решённых задач. Наиболее известной открытой проблемой является вопрос о соотношении классов **P** и **NP**, которая включена в список из 7 задач тысячелетия, за решение которых обещан приз в 1 миллион долларов. Изучению этой проблемы будут посвящены первые несколько лекций.

3 Основные темы курса

1. **О проблеме $\mathbf{P} \stackrel{?}{=} \mathbf{NP}$.** Неформальное определение классов **P** и **NP**, примеры задач из двух классов. Постановка вопроса о равенстве классов, математические и возможные общественные следствия из $\mathbf{P} = \mathbf{NP}$ и из $\mathbf{P} \neq \mathbf{NP}$. История исследования проблемы и известные препятствия к её решению.

2. **Модели вычислений.** Виды вычислительных ресурсов. Модели вычислений: одноленточные и многоленточные машины Тьюринга, обзор других подходов. Измерение времени работы алгоритма и используемой памяти. Асимптотические оценки. Определение сложности задачи в худшем случае. Тезис Чёрча–Тьюринга в сильной форме.

3. **Временна́я сложность.** Функции, конструируемые по времени. Сложностные классы $\mathbf{DTIME}(T(n))$, **P**, **E**, **EXP** и др. Неконструктивные оценки сложности для принадлежности к **P**. Два определения класса **NP** и их эквивалентность. Класс **coNP**. Класс **NEXP**. Полиномиальная сводимость по Карпу и её свойства. **NP**-трудные и **NP**-полные задачи. Теорема Кука–Левина. Примеры **NP**-полных задач: выполнимость 3-КНФ, клика, вершинное покрытие, гамильтонов путь, задача о рюкзаке и др. Задачи поиска и их полиномиальная разрешимость в случае $\mathbf{P} = \mathbf{NP}$. Теорема об иерархии по времени. Теорема Ладнера о существовании **NP**-промежуточных задач. Теорема Бейкера–Джилла–Соловея о нерелятивизуемости утверждения $\mathbf{P} = \mathbf{NP}$.

4. **Полиномиальная иерархия сложностных классов.** Классы Σ_n^p и Π_n^p. Полные задачи на уровнях иерархии. Коллапсирование полиномиальной иерархии в случае $\Sigma_n^p = \Pi_n^p$ и в случае существования **PH**-полных задач. Игровая интерпретация классов полиномиальной иерархии. Интерпретация через альтернирующие машины. Интерпретация через вычисления с оракулом.

5. **Пространственная сложность.** Функции, конструируемые по памяти. Классы $\mathbf{DSPACE}(S(n))$, $\mathbf{NSPACE}(S(n))$, **PSPACE**. Теорема Сэвича о моделировании

недетерминированных вычислений детерминированными с квадратичным увеличением памяти. **PSPACE** = **NPSPACE**. **PSPACE**-полнота. Примеры **PSPACE**-полных задач. Интерпретация **PSPACE** через игры и через альтернирующие вычисления. Классы **L**, **NL** и **coNL**. **NL**-полнота. Примеры **NL**-полных задач. Теорема Иммермана–Селепченьи: **NL** = **coNL**.

6. **Схемная сложность.** Схемы из функциональных элементов. Схемная сложность и класс $\mathbf{P}_{/\mathrm{poly}}$. Связь с прямолинейными программами и с машинами Тьюринга, принимающими подсказки. Теорема Карпа–Липтона (если $\mathbf{NP} \subset \mathbf{P}_{/\mathrm{poly}}$, то $\mathbf{PH} = \Sigma_2^p$). Теорема Мейера (если $\mathbf{EXP} \subset \mathbf{P}_{/\mathrm{poly}}$, то $\mathbf{EXP} = \Sigma_2^p$). Глубина схемы. Классы $\mathbf{NC}^k$ и $\mathbf{AC}^k$. Эффективные сумматоры и мультипликаторы. Нижние оценки схемной сложности.

7. **Вероятностные алгоритмы.** Рандомизированные машины Тьюринга. Классы **BPP**, **RP**, **coRP** и **ZPP**, соотношения между ними. Устойчивость определений к точному выбору констант. Вложения **BPP** в $\mathbf{P}_{/\mathrm{poly}}$ и в $\Sigma_2^p \cap \Pi_2^p$ (теоремы Эйдлмана и Гача–Сипсера). Вероятностные вычисления на логарифмической памяти. Вероятностный алгоритм проверки достижимости в неориентированном графе.

8. **Дерандомизация**. Обзор методов преобразования вероятностного алгоритма в детерминированный или уменьшения числа использованных в алгоритме случайных битов. Методы условных математических ожиданий и попарно независимых случайных величин. Обзор псевдослучайных конструкций.

9. **Сложность задач подсчёта**. Задачи подсчёта и их связь с задачами распознавания. Класс #**P**. Полиномиальная сводимость для задач подсчёта и #**P**-полные задачи. Класс **PP**. Если **PP** = **P**, то #**P** = **FP**. Теорема Тоды: полиномиальная иерархия вложена в $\mathbf{P}^{\#\mathsf{SAT}}$.

10. **Сложность в среднем**. Различные подходы к измерению сложности в среднем. Классы **distP** и **distNP**. Сводимость в среднем и **distNP**-полные задачи.

11. **Дополнительные темы**. Если останется время, можно изучить дополнительные темы, либо заменить на них некоторые из последних. Примеры возможных тем: интерактивные доказательства, вероятностно проверяемые доказательства, основания криптографии, тотальные задачи поиска.

4 Выставление оценок

Отчётность по курсу — экзамен с оценкой по 10-балльной шкале. Оценка ставится за 3 компонента: устный экзамен (оценка E), контрольные и домашние работы (оценка C), а также индивидуальный проект (оценка P). Индивидуальный проект оценивается по шкале от 0 до 12 и идёт в итоговую оценку с весом 0.2. Точная функция перевода оценки C в итоговую будет объявлена не позднее начала зачётной недели и может быть нелинейной, после чего можно будет досдавать домашние задачи до желаемой оценки. Оценки C и P должны быть зафиксированы до начала устного экзамена, после чего оценка E идёт в итоговую с весом 0.5. Итоговая оценка округляется сначала к ближайшему числу до десятых долей балла, а затем вниз до целого (т. е. дробная часть

от 0 до 0.95 округляется вниз, а от 0.95 до 1 вверх). Условия получения различных оценок:

- Для получения зачёта по курсу нужно одновременно получить хотя бы 3 в итоговой оценке и за экзамен.
- Для получения оценки «отл» требуется сдать проект хотя бы на 5 баллов.
- Для получения оценки «отл(10)» требуется сдать экзамен хотя бы на 8 баллов.

Ниже написаны более подробно правила выставления оценок по трём компонентам.

4.1 Контрольные и домашние задания

Планируется провести 3 контрольных за семестр. Примерные даты: 11 октября (во время лекции), середина ноября и зачётная неделя. На контрольной будет дано 6 задач, оцениваемые от 0 до 10 баллов. Отличным результатом считается решение 4 задач. Если задача решена менее, чем на 8 баллов, похожая задача будет выдана в домашней работе исходя из 5 баллов. Баллы за домашнюю задачу добавляются к баллам за контрольную, но сумма не может превысить 8. Также в домашнюю работу могут быть включены другие задачи, оцениваемые в 10 баллов.

Если контрольная пропускается по уважительной причине, об этом нужно заявить до начала контрольной. В этом случае задачи из домашнего задания оцениваются из 8 баллов.

4.2 Проект

Индивидальный проект сдаётся в виде текста. Проект может быть трёх видов: реферативным, исследовательским или программистским. Реферативный проект означает, что студент разбирается в какой-то области или теореме и пишет про неё связный самодостаточный текст на русском языке. Исследовательский проект означает изучение какой-то конкретной задачи с точки зрения сложностных классов и написание отчёта об исследовании. Программистский проект предполагает реализацию какого-либо алгоритма и написание отчёта по его работе. Например, это может быть приближённый, эвристический или нетривиальный детерминированный алгоритм для решения какой-либо **NP**-полной задачи. Однако оценивается прежде всего текст отчёта, а не программа. Более подробные критерии и список возможных тем для проектов будут опубликованы дополнительно. До 15 октября нужно будет выбрать тему, до 19 ноября прислать аннотацию и план текста, до 3 декабря — предварительный текст, до 16 декабря — окончательный. (Даты могут меняться в зависимости от расписания экзаменов).

4.3 Экзамен

До получения билета на устном экзамене нужно зафиксировать оценки за контрольные, домашние задания и проект. В билете будет 3 вопроса: 2 теоретических и задача. Каждый из вопросов оценивается в 3 балла, ещё 1 балл ставится за общее впечатление или ответы на дополнительные вопросы. В первом теоретическом вопросе будет дан список сложностных классов, про которые нужно рассказать известные вам соотношения при различных теоретико-сложностных предположениях. Во втором вопросе

нужно будет доказать некоторую теорему. Задача будет дана из заранее известного списка. При подготовке ответа можно пользоваться любыми материалами, однако отвечать нужно с чистого листа. Более подробные правила будут опубликованы в конце семестра вместе с программой экзамена.

5 Литература

5.1 Основная литература

1. Д. В. Мусатов, «Сложность вычислений: классика и современность», черновик книги (обновляется в телеграм-канале)

2. S. Arora, B. Barak, "Computational Complexity: A Modern Approach", Cambridge University Press, 2009 (Черновики доступны по адресу `http://www.cs.princeton.edu/theory/index.php/Compbook/Draft`)

3. M. Sipser, "Introduction to the Theory of Computation", Course Technology, 2005

5.2 Дополнительная литература

4. C. Moore, S. Mertens, "The Nature of Computation", Oxford University Press, 2011

5. O. Goldreich, "Computational Complexity: A Conceptual Perspective", Cambridge University Press, 2008

6. J. Katz, "Notes on Complexity Theory", 2011
`http://www.cs.umd.edu/~jkatz/complexity/f11/all.pdf`

7. C. Papadimitriou, "Computational Complexity", Addison Wesley, 1994

8. В. Н. Крупский. "Введение в сложность вычислений", Москва, Факториал Пресс, 2006

9. С. А. Абрамов, "Лекции о сложности алгоритмов", Москва, МЦНМО, 2009

10. М. Гэри, Д. Джонсон, "Вычислительные машины и труднорешаемые задачи", Москва, Мир, 1982

11. М. Вялый, А. Китаев, А. Шень, "Классические и квантовые вычисления", Москва, МЦНМО, 1999,
`http://www.mccme.ru/free-books/qcomp/qps00205.zip`

5.3 Полезные сетевые ресурсы

1. «Зоопарк» сложностных классов.
`https://complexityzoo.uwaterloo.ca`

2. Dick Lipton, "Gödel's Lost Letter and P=NP",
`http://rjlipton.wordpress.com/`

3. Scott Aaronson, “Shtetl-Optimized”,
   http://www.scottaaronson.com/blog/

4. Computational Complexity Blog,
   http://blog.computationalcomplexity.org/

5. Вопросы и ответы по теоретической информатике
   http://cs.stackexchange.com/ (вопросы учебного уровня)
   http://cstheory.stackexchange.com/ (вопросы исследовательского уровня)