

# Программа по информатике для перевода и восстановления

## на 2 семестр

1. Алгоритм. Модель вычислений (пример: одноленточные машины Тьюринга). Алгоритм. Сложность алгоритма (время, используемая память). Рекурсия.
2. Сортировка, постановка задачи. Алгоритмы сортировки: пузырьком, вставками, Шелла. Сложность алгоритмов в худшем случае и в среднем.
3. Пирамидальная сортировка. Сложность алгоритмов в худшем случае и в среднем.
4. Сортировка слиянием. Сложность алгоритмов в худшем случае и в среднем.
5. Поразрядная сортировка. Сложность алгоритмов в худшем случае и в среднем.
6. Структуры данных. Операции, реализуемые в структурах данных, их сложности. Абстрактный тип данных
7. Массив, список, односвязный список, кольцевой список, стек.
8. Очередь. Очередь с приоритетами.
9. Куча.
10. Объектно-ориентированное программирование. Язык C++.
11. Метод динамического программирования. Пример задачи, которая решается методом динамического программирования. Оценки сложности.

## на 3 семестр

1. Алгоритм. Сложность алгоритма (время, используемая память). Рекурсия.
2. Сортировка, постановка задачи. Алгоритмы сортировки: пузырьком, вставками. Пирамидальная сортировка. Сортировка слиянием. Быстрая сортировка. Поразрядные сортировки. Сложность алгоритмов в худшем случае и в среднем.
3. Структуры данных. Операции, реализуемые в структурах данных, их сложности. Абстрактный тип данных.
4. Массив, список, односвязный список, кольцевой список. Стек, очередь, дек.
5. Двоичная куча. Очередь с приоритетом.
6. Деревья поиска. Декартовы деревья. AVL-деревья. Красно-черные деревья. Деревья по неявному ключу.
7. Хеш-таблица. Разрешение коллизий методом цепочек. Разрешение конфликтов методом открытой адресации.
8. Задача RMQ. Sparse table. Дерево отрезков.
9. Задача LCA. Метод двоичного подъема. Сведение RMQ к LCA и наоборот.
10. Граф. Ориентированный граф. Представления графа. Обход графа в глубину и в ширину. Топологическая сортировка. Подсчет числа путей в орграфе.
11. Сильно связанные компоненты. Алгоритм Косарайю. Алгоритм Тарьяна.
12. Поиск кратчайших путей в графе. Алгоритм Дейкстры. Алгоритм Форда-Беллмана. Алгоритм Флойда. Алгоритм A\*. Эвристики.
13. Минимальное остовное дерево. Алгоритм Прима.
14. Система непересекающихся множеств. Алгоритм Крускала.
15. Алгоритм Борувки.

16. Потоки, алгоритм Форда-Фалкерсона. Алгоритм Эдмондса-Карпа. Алгоритм Диница.
17. Основные понятия ООП. Конструкторы/деструкторы. Перегрузка методов. Соккрытие методов. Какие методы и операторы необходимы для использования типа в качестве параметра стандартного шаблонного контейнера? Ключевые слова `virtual` и `const`. Что такое срезка? Множественное наследование.
18. Исключения C++. Генерирование и перехват исключений. Обработка ошибок в конструкторах и деструкторах.
19. Шаблоны C++.
20. Устройство контейнеров STL, основные операции и их стоимость, особенности использования: `vector`, `list`, `deque`, `stack`, `map`, `set`, `bitset` и `vector<bool>`, `unordered_map`, `priority_queue`.
21. STL: итераторы. Что такое итератор? Какие бывают итераторы? Что такое итератор с произвольным доступом?
22. Сортировка и поиск в STL. В каких контейнерах элементы хранятся упорядоченно? Куча в STL. Ассоциативный массив. Интерфейс, варианты реализации - хэш-таблицы, `rb-tree`.

## на 4 семестр

1. Алгоритм. Модель вычислений (пример: одноленточные машины Тьюринга). Алгоритм. Сложность алгоритма (время, используемая память). Рекурсия.
2. Сортировка, постановка задачи. Алгоритмы сортировки: пузырьком, вставками. Пирамидальная сортировка. Поразрядная сортировка. Сортировка слиянием. Сложность алгоритмов в худшем случае и в среднем.
3. Структуры данных. Операции, реализуемые в структурах данных, их сложности. Абстрактный тип данных.
4. Массив, список, односвязный список, кольцевой список, стек. Очередь. Очередь с приоритетами.
5. Объектно-ориентированное программирование. Язык C++.
6. Граф. Ориентированный граф. Представления графа. Обход графа в глубину и в ширину. Топологическая сортировка. Подсчет числа путей в орграфе.
7. Сильно связанные компоненты. Алгоритм Тарьяна.
8. Поиск кратчайших путей в графе. Алгоритм Флойда. Алгоритм Дейкстры. Алгоритм Форда-Беллмана. Алгоритм A\*. Эвристики.
9. Минимальное остовное дерево. Алгоритм Прима. Биномиальная куча. Амортизационная стоимость. Фибоначчиева куча.
10. Система непересекающихся множеств. Алгоритм Крускала.
11. Потоки, алгоритм Форда-Фалкерсона.
12. Деревья поиска. Декартовы деревья. Дерево Фенвика. Дерево отрезков и динамическое программирование (Sparse table) для RMQ. Сведение RMQ к LCA и наоборот. Поиск нескольких минимумов на отрезке.
13. Задача поиска подстрок. Алгоритм Рабина-Карпа. Конечный автомат. Алгоритм Бойера-Мура. Алгоритм Кнута-Морриса-Пратта.
14. Алгоритм Ахо-Корасика. Суффиксное дерево. Бор. Алгоритм Укконена. Суффиксный массив.

15. Основные понятия ООП. Конструкторы/деструкторы. Перегрузка методов. Соккрытие методов. Какие методы и операторы необходимы для использования типа в качестве параметра стандартного шаблонного контейнера? Ключевые слова `virtual` и `const`. Что такое срезка? Множественное наследование.
16. Исключения. Генерирование и перехват исключений. `throw`-списки. Их изменение в переопределенных методах. Обработка ошибок в конструкторах и деструкторах.
17. Шаблоны. Последовательные контейнеры STL и адаптеры. Что такое адаптер над контейнером STL?
18. STL: итераторы. Что такое итератор? Какие бывают итераторы? Что такое итератор с произвольным доступом?
19. Устройство, основные операции и их стоимость, особенности использования для: `vector`, `list`, `deque`, `stack`, `map`, `set`, `bitset` и `vector<bool>`, `unordered_map`, `priority_queue`,
20. Сортировка и поиск в STL. В каких контейнерах элементы хранятся упорядоченно? Куча в STL. Ассоциативный массив. Интерфейс, варианты реализации - хэш-таблицы, `rb-tree`. Реализации в языке.

## на 5 семестр

1. Алгоритм. Сложность алгоритма (время, используемая память). Рекурсия.
2. Сортировка, постановка задачи. Алгоритмы сортировки: пузырьком, вставками. Пирамидальная сортировка. Сортировка слиянием. Быстрая сортировка. Поразрядные сортировки. Сложность алгоритмов в худшем случае и в среднем.
3. Структуры данных. Операции, реализуемые в структурах данных, их сложности. Абстрактный тип данных.
4. Массив, список, односвязный список, кольцевой список. Стек, очередь, дек.
5. Двоичная куча. Очередь с приоритетом.
6. Деревья поиска. Декартовы деревья. АВЛ-деревья. Красно-черные деревья. Деревья по неявному ключу.
7. Хэш-таблица. Разрешение коллизий методом цепочек. Разрешение конфликтов методом открытой адресации.
8. Задача RMQ. `Sparse table`. Дерево отрезков.
9. Задача LCA. Метод двоичного подъема. Сведение RMQ к LCA и наоборот.
10. Граф. Ориентированный граф. Представления графа. Обход графа в глубину и в ширину. Топологическая сортировка. Подсчет числа путей в орграфе.
11. Сильно связанные компоненты. Алгоритм Косараяю. Алгоритм Тарьяна.
12. Поиск кратчайших путей в графе. Алгоритм Дейкстры. Алгоритм Форда-Беллмана. Алгоритм Флойда. Алгоритм A\*. Эвристики.
13. Минимальное остовное дерево. Алгоритм Прима.
14. Система непересекающихся множеств. Алгоритм Крускала.
15. Алгоритм Борувки.
16. Потоки, алгоритм Форда-Фалкерсона. Алгоритм Эдмондса-Карпа. Алгоритм Диница.
17. Задача поиска подстрок. Префикс-функция, Z-функция. Алгоритм Кнута-Морриса-Пратта.
18. Алгоритм Ахо-Корасика.
19. Суффиксный массив.
20. Бор. Суффиксное дерево. Алгоритм Укконена.

21. Длинная арифметика. Алгоритм Карацубы.
22. Быстрое преобразование Фурье.
23. Основные понятия ООП. Конструкторы/деструкторы. Перегрузка методов. Соккрытие методов. Какие методы и операторы необходимы для использования типа в качестве параметра стандартного шаблонного контейнера? Ключевые слова `virtual` и `const`. Что такое срезка? Множественное наследование.
24. Исключения C++. Генерирование и перехват исключений. Обработка ошибок в конструкторах и деструкторах.
25. Шаблоны C++.
26. Устройство контейнеров STL, основные операции и их стоимость, особенности использования: `vector`, `list`, `deque`, `stack`, `map`, `set`, `bitset` и `vector<bool>`, `unordered_map`, `priority_queue`.
27. STL: итераторы. Что такое итератор? Какие бывают итераторы? Что такое итератор с произвольным доступом?
28. Сортировка и поиск в STL. В каких контейнерах элементы хранятся упорядоченно? Куча в STL. Ассоциативный массив. Интерфейс, варианты реализации - хэш-таблицы, `rb-tree`.
29. Синхронизация: условные переменные, мьютексы и семафоры.

## на 6 семестр

1. Алгоритм. Модель вычислений (пример: одноленточные машины Тьюринга). Алгоритм. Сложность алгоритма (время, используемая память). Рекурсия.
2. Сортировка, постановка задачи. Алгоритмы сортировки: пузырьком, вставками. Пирамидальная сортировка. Поразрядная сортировка. Сортировка слиянием. Сложность алгоритмов в худшем случае и в среднем.
3. Структуры данных. Операции, реализуемые в структурах данных, их сложности. Абстрактный тип данных.
4. Массив, список, односвязный список, кольцевой список, стек. Очередь. Очередь с приоритетами.
5. Объектно-ориентированное программирование. Язык C++.
6. Граф. Ориентированный граф. Представления графа. Обход графа в глубину и в ширину. Топологическая сортировка. Подсчет числа путей в орграфе.
7. Сильно связанные компоненты. Алгоритм Тарьяна.
8. Поиск кратчайших путей в графе. Алгоритм Флойда. Алгоритм Дейкстры. Алгоритм Форда-Беллмана. Алгоритм A\*. Эвристики.
9. Минимальное остовное дерево. Алгоритм Прима. Биномиальная куча. Амортизационная стоимость. Фибоначчиева куча.
10. Система непересекающихся множеств. Алгоритм Крускала.
11. Потoki, алгоритм Форда-Фалкерсона.
12. Деревья поиска. Декартовы деревья. Дерево Фенвика. Дерево отрезков и динамическое программирование (Sparse table) для RMQ. Сведение RMQ к LCA и наоборот. Поиск нескольких минимумов на отрезке.
13. Задача поиска подстрок. Алгоритм Рабина-Карпа. Конечный автомат. Алгоритм Бойера-Мура. Алгоритм Кнута-Морриса-Пратта.

14. Алгоритм Ахо-Корасика. Суффиксное дерево. Бор. Алгоритм Укконена. Суффиксный массив.
15. Основные понятия ООП. Конструкторы/деструкторы. Перегрузка методов. Соккрытие методов. Какие методы и операторы необходимы для использования типа в качестве параметра стандартного шаблонного контейнера? Ключевые слова `virtual` и `const`. Что такое срезка? Множественное наследование.
16. Исключения. Генерирование и перехват исключений. `throw`-списки. Их изменение в переопределенных методах. Обработка ошибок в конструкторах и деструкторах.
17. Шаблоны. Последовательные контейнеры STL и адаптеры. Что такое адаптер над контейнером STL?
18. STL: итераторы. Что такое итератор? Какие бывают итераторы? Что такое итератор с произвольным доступом?
19. Устройство, основные операции и их стоимость, особенности использования для: `vector`, `list`, `deque`, `stack`, `map`, `set`, `bitset` и `vector<bool>`, `unordered_map`, `priority_queue`,
20. Сортировка и поиск в STL. В каких контейнерах элементы хранятся упорядоченно? Куча в STL. Ассоциативный массив. Интерфейс, варианты реализации - хэш-таблицы, `rb-tree`. Реализации в языке.

## на 7 семестр

1. Алгоритм. Сложность алгоритма (время, используемая память). Рекурсия.
2. Сортировка, постановка задачи. Алгоритмы сортировки: пузырьком, вставками. Пирамидальная сортировка. Сортировка слиянием. Быстрая сортировка. Поразрядные сортировки. Сложность алгоритмов в худшем случае и в среднем.
3. Структуры данных. Операции, реализуемые в структурах данных, их сложности. Абстрактный тип данных.
4. Массив, список, односвязный список, кольцевой список. Стек, очередь, дек.
5. Двоичная куча. Очередь с приоритетом.
6. Деревья поиска. Декартовы деревья. AVL-деревья. Красно-черные деревья. Деревья по неявному ключу.
7. Хэш-таблица. Разрешение коллизий методом цепочек. Разрешение конфликтов методом открытой адресации.
8. Задача RMQ. Sparse table. Дерево отрезков.
9. Задача LCA. Метод двоичного подъема. Сведение RMQ к LCA и наоборот.
10. Граф. Ориентированный граф. Представления графа. Обход графа в глубину и в ширину. Топологическая сортировка. Подсчет числа путей в орграфе.
11. Сильно связные компоненты. Алгоритм Косарайю. Алгоритм Тарьяна.
12. Поиск кратчайших путей в графе. Алгоритм Дейкстры. Алгоритм Форда-Беллмана. Алгоритм Флойда. Алгоритм A\*. Эвристики.
13. Минимальное остовное дерево. Алгоритм Прима.
14. Система непересекающихся множеств. Алгоритм Крускала.
15. Алгоритм Борувки.
16. Потоки, алгоритм Форда-Фалкерсона. Алгоритм Эдмондса-Карпа. Алгоритм Диница.
17. Задача поиска подстрок. Префикс-функция, Z-функция. Алгоритм Кнута-Морриса-Пратта.
18. Алгоритм Ахо-Корасика.

19. Суффиксный массив.
20. Бор. Суффиксное дерево. Алгоритм Укконена.
21. Длинная арифметика. Алгоритм Карацубы.
22. Быстрое преобразование Фурье.
23. Основные понятия ООП. Конструкторы/деструкторы. Перегрузка методов. Скрытие методов. Какие методы и операторы необходимы для использования типа в качестве параметра стандартного шаблонного контейнера? Ключевые слова `virtual` и `const`. Что такое срезка? Множественное наследование.
24. Исключения C++. Генерирование и перехват исключений. Обработка ошибок в конструкторах и деструкторах.
25. Шаблоны C++.
26. Устройство контейнеров STL, основные операции и их стоимость, особенности использования: `vector`, `list`, `deque`, `stack`, `map`, `set`, `bitset` и `vector<bool>`, `unordered_map`, `priority_queue`.
27. STL: итераторы. Что такое итератор? Какие бывают итераторы? Что такое итератор с произвольным доступом?
28. Сортировка и поиск в STL. В каких контейнерах элементы хранятся упорядоченно? Куча в STL. Ассоциативный массив. Интерфейс, варианты реализации - хэш-таблицы, `rb-tree`.
29. Синхронизация: условные переменные, мьютексы и семафоры.
30. Байесовские методы классификации. Наивный байесовский классификатор.
31. Логистическая регрессия. L1 и L2 регуляризации. Теорема об оптимальности.
32. Метод опорных векторов, решение двойственной задачи. Спрямоугоющее пространство. Ядра.

## на 8 семестр

### Алгоритмы и структуры данных

1. Быстрая сортировка (QuickSort).
2. Сортировка слиянием (MergeSort).
3. Двоичная куча и сортировка кучей (HeapSort).
4. Хэш-таблица, полиномиальная хэш-функция.
5. Динамическое программирование: общая идея, линейная динамика, матричная динамика
6. на отрезках.
7. Амортизационный анализ.
8. RMQ.
9. LCA: сведение к RMQ и метод двоичного подъема.
10. Декартово дерево. Декартово дерево по неявному ключу.
11. Минимальное остовное дерево: алгоритмы Прима и Крускала.
12. Максимальные потоки в сети. Методы: Форда-Фалкерсона; Эдмондса-Карпа (б/д).
13. Обход графа в глубину, ширину.
14. Поиск кратчайших путей в графе: алгоритмы Дейкстры, Форда-Беллмана, Флойда-Уоршелла.
15. Поиск сильносвязных компонент в графе.

16. Мосты и точки сочленения в графе.
17. Нахождение подстроки в строке: префикс-функция, алгоритм Кнута-Морриса-Пратта.
18. Стандартные контейнеры: `vector`, `deque`, `queue`, `priority_queue`, `set`, `map`, итераторы, компараторы.

#### Машинное обучение

1. Байесовские методы классификации. Наивный байесовский классификатор.
2. Логистическая регрессия. L1 и L2 регуляризации. Теорема об оптимальности.
3. Метод опорных векторов, решение двойственной задачи. Спрямяющее пространство. Ядра.
4. Многомерная линейная регрессия. Лассо тибширани. Гребневая регрессия.
5. Бустинг. Алгоритм AdaBoost

#### Формальные языки и трансляции

1. Недетерминированные конечные автоматы. Различные варианты определения. Детерминированные конечные автоматы. Их эквивалентность.
2. Регулярные выражения. Теорема Клини об эквивалентности регулярных выражений и конечных автоматов.
3. Минимизация конечных автоматов. Алгоритм минимизации. Алгоритм проверки эквивалентности регулярных выражений.
4. Порождающие грамматики. Иерархия Хомского. Праволинейные, контекстно-свободные, контекстно-зависимые грамматики (определения). Эквивалентность праволинейных грамматик и конечных автоматов.
5. Контекстно-свободные грамматики. Нормальная форма Хомского для контекстно-свободных грамматик.
6. Автоматы с магазинной памятью. Варианты определения. Эквивалентность автоматов с магазинной памятью и контекстно-свободных грамматик.
7. Леммы о разрастании для автоматных и контекстно-свободных языков. Примеры языков, не лежащих в данных классах.
8. Алгоритмы синтаксического разбора для контекстно-свободных грамматик. Алгоритмы Кока-Янгера-Касами и Эрли.