

Индивидуальный предприниматель Майоров Рустем Марванович

Утверждено приказом
№01-02/17 от 25.06.2025г.

Рассмотрено
педагогическим советом
ЦРШ «Код Успеха»
Протокол №2 от 02.06.2025г.

Дополнительная общеобразовательная общеразвивающая программа
технической направленности
«Алгоритмы и структуры данных. С++. Уровень 3»

Возраст обучающихся: 13-17
Срок реализации: 1 год
Форма обучения – очная
Уровень программы - продвинутый
Код программы - 20.3

Автор-составитель:
Горшков Даниил Александрович

г. Ульяновск, 2025

Оглавление

1	Пояснительная записка	2
2	Календарный учебный график	8
3	Учебный план	8
4	Содержание программы	10
5	Оценочные и методические материалы	21
6	Материально-техническое обеспечение программы	23
7	Кадровое обеспечение	23
8	Список литературы	23

1 Пояснительная записка

Дополнительная общеобразовательная программа «Алгоритмы и структуры данных. С++. Уровень 3» разработана согласно требованиям следующих нормативных документов:

- Федеральный Закон «Об образовании в Российской Федерации» от 29.12.2012 № 273-ФЗ.
- Концепция развития дополнительного образования детей (утверждена распоряжением Правительства РФ от 04.09.2014 № 1726-р).
- СП 2.4.3648-20 "Санитарно-эпидемиологические требования к организациям воспитания и обучения, отдыха и оздоровления детей и молодежи"
- СанПиН 1.2.3685-21 "Гигиенические нормативы и требования к обеспечению безопасности и (или) безвредности для человека факторов среды обитания"
- Порядок организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам (утвержден приказом Министерства просвещения РФ от 9 ноября 2018 г. N 196).
- Стратегия развития воспитания в Российской Федерации на период до 2025 года (утверждена Распоряжением Правительства Российской Федерации от 29 мая 2015 г. N 996-р)

1.1. Направленность

Направленность дополнительной образовательной программы «Алгоритмы и структуры данных. С++. Уровень 3» – научно-техническая. Она является прикладной, носит практико-ориентированный характер и направлена на овладение воспитанниками продвинутых алгоритмов и структур данных. Обучение по данной программе создаёт благоприятные условия для интеллектуального и духовного воспитания личности ребенка, социально - культурного и профессионального самоопределения, развития познавательной активности и творческой самореализации учащихся.

1.2. Актуальность программы

Данная дополнительная общеобразовательная программа соотносится с тенденциями развития дополнительного образования и согласно Концепции развития дополнительного образования способствует:

- созданию необходимых условий для личностного развития обучающихся, позитивной социализации и профессионального самоопределения;
- удовлетворению индивидуальных потребностей обучающихся в интеллектуальном развитии, а также в занятиях научно-техническим творчеством;
- формированию и развитию творческих способностей учащихся, выявлению, развитию и поддержке талантливых учащихся;
- обеспечению духовно-нравственного, гражданского, патриотического, трудового воспитания учащихся;

Кроме того, актуальность программы обуславливается тем, что при обычном обучении информатике, алгоритмы и структуры данных не изучаются, это не способствует успешному выступлению на олимпиадах по информатике. Обучение по данной программе создает благоприятные условия для более глубокого «погружения» учащихся в мир логики, математического моделирования, для интеллектуального и духовного воспитания личности ребенка, социально - культурного и профессионального самоопределения, развития познавательной активности и творческой самореализации учащихся. Она рассчитана на сотворчество и сотрудничество педагога и воспитанников. Данная программа дает возможность детям творчески мыслить, находить самостоятельные индивидуальные решения, а полученные умения и навыки применять в жизни. Развитие творческих способностей помогает также в профессиональной ориентации подростков.

1.3. Педагогическая целесообразность

Педагогическая целесообразность заключается в том, что сделанный в представленной программе акцент на практическую работу позволяет повысить результативность обучения информатике и ИКТ при совместном преподавании школьного основного курса и данного дополнительного, расширить мировоззрение учащихся, повысить предметные и межпредметные ЗУН и УУН, успешно освоить учебный материал и участвовать в олимпиадах, осознанно выбрать профиль дальнейшего обучения и будущей профессии.

1.4. Цель и задачи программы

Цель программы: развитие творческих способностей учащихся, овладение теоретическими и практическими навыками решения алгоритмических задач повышенного уровня сложности программными средствами.

Задачи:

образовательные:

обучить продвинутым алгоритмам и способам их реализации с помощью языка программирования С++;

обучить структурам данных, способам их реализации с помощью языка программирования С++ и использованию соответствующих классов из стандартных библиотек;

обучить продвинутым алгоритмам и способам их реализации с помощью языка программирования С++;

обучить продвинутым приемам работы в интегрированных средах программирования;

совершенствовать навыки тестирования, поиска и исправления ошибок в программном коде;

воспитательные:

воспитывать культуру написания «чистого», понятного для других программистов, кода, соблюдения рекомендаций по выбору имен для идентификаторов;

воспитать культуру проектной деятельности, в том числе умение планировать, работать в коллективе; чувство ответственности за результаты своего труда, используемого другими людьми;

формировать установку на позитивную социальную деятельность в информационном обществе, недопустимость действий, нарушающих правовые и этические нормы работы с информацией;

содействовать повышению привлекательности науки, научно-технического творчества для подрастающего поколения;

формировать у подрастающего поколения ответственное отношение к своему здоровью и потребности в здоровом образе жизни, соблюдении правил гигиены;

развивающие:

быстро ориентироваться в динамично развивающемся и обновляющемся информационном пространстве; получать, использовать и создавать разнообразную информацию; принимать обоснованные решения и решать проблемы на основе полученных знаний, умений и навыков;

овладение умениями строить математические объекты, в том числе алгоритмы, структуры данных, логические формулы и программы на формальном языке, удовлетворяющие заданному описанию; использовать общепользовательские инструменты и настраивать их для нужд пользователя;

развитие алгоритмического мышления, способностей к формализации, элементов системного мышления;

приобретение опыта создания, редактирования, оформления, сохранения, передачи информационных объектов различного типа с помощью современных программных средств; построение компьютерных моделей, коллективной реализации информационных проектов, преодоление трудностей в процессе интеллектуального проектирования, информационной деятельности в различных сферах, востребованных на рынке труда; решения сложных задач и олимпиадных задач программирования.

1.5. Планируемые результаты

Планируемые результаты обучения по дополнительной общеобразовательной программе «Алгоритмы и структуры данных. С++. Уровень 3»:

Предметные:

- умение анализировать поставленные задачи и создавать соответствующие информационные модели;
- умение применять алгоритмы, комбинировать и модифицировать их для решения поставленных задач, выбирать оптимальные структуры данных;
- умение применять основные конструкции языка С++ для реализации алгоритмов решения поставленных задач;

- умение тестировать и выполнять отладку программного кода, применять инструментальные средства отладки.

Личностные:

- чувство патриотизма, уважения к своему народу, гордости за свой край;
- ответственное отношение к труду, качеству своей деятельности;
- овладение основами саморазвития и самовоспитания в соответствии с общечеловеческими ценностями и идеалами гражданского общества; готовность и способность к самостоятельной, творческой и ответственной деятельности;
- умение общаться и сотрудничать со сверстниками, детьми старшего и младшего возраста, взрослыми в процессе образовательной, общественно полезной, творческой и других видов деятельности;
- навыки сотрудничества с взрослыми и сверстниками в разных социальных ситуациях, умение не создавать конфликтов и находить выходы из спорных ситуаций;
- сформированность познавательных интересов и творческая активность в данной области.

Метапредметные:

- умение определять и формулировать цель деятельности на занятии, проговаривать последовательность действий с помощью педагога;
- умение планировать, контролировать и оценивать свои действия в соответствии с поставленной задачей и условиями её реализации, определять наиболее эффективные способы достижения результата;
- способность самостоятельно оценивать и принимать решения, определяющие стратегию поведения, с учетом гражданских и нравственных ценностей;
- умение понимать причины успеха/неуспеха учебной деятельности и способности конструктивно действовать даже в ситуациях неуспеха.

В результате обучения выпускники должны знать:

- алгоритмы, их вычислительную сложность;
- основные структуры данных и вычислительную сложность выполнения операций на этих структурах;
- стандартные библиотечные функции, реализующие изученные алгоритмы;
- стандартные классы, реализующие изученные структуры данных.

должны уметь:

- оценивать вычислительную сложность алгоритмов решения задач;
- применять изученные алгоритмы для решения задач;
- составлять программный код, реализующий тот или иной алгоритм;
- определять необходимые для решения задачи структуры данных;
- оформлять программный код с соответствием с общепринятыми правилами;

- осуществлять поиск исходных данных, приводящих к неверному результату, определять причину неверного результата и принимать решение по оптимальному исправлению ошибок в программном коде;
- работать в команде над решением задачи;
- проявлять чувство коллективизма, взаимопомощи, взаимопонимания;
- уважать труд своих товарищей;
- стремиться к совершенствованию своих навыков.

1.6. Отличительные особенности программы

Отличительные особенности данной дополнительной программы от существующих образовательных программ в том, что она содержит дополнительный изучаемый материал, значительно расширяет возможности формирования универсальных учебных и предметных навыков. В программе большое внимание уделяется практическим заданиям. Проверка заданий осуществляется комбинированным способом. На первом этапе автоматически с помощью тестирующей системы. На этом этапе отсекается большая часть неверных решений. Второй этап контроля – проверка отправленного кода преподавателем. На этом этапе проверяется, выполнены ли ограничения по используемым конструкциям языка, оптимальность решения и правильность оформления кода. Допущенные ошибки разбираются на занятии. Такой подход приучает детей к самостоятельной работе, освобождает преподавателя от рутины проверки заведомо неверных решений, тем самым повышая эффективность учебного процесса.

Все учебные материалы представлены в формате презентаций и/или лекций. Презентации удобно использовать с помощью мультимедийного проектора. Обучаемые имеют к презентациям и/или лекциям постоянный доступ в сети Интернет и могут использовать их в качестве опорного конспекта.

1.7. Условия реализации программы

В освоении настоящей учебной программы участвуют дети в возрасте 13-17 лет. Условием отбора в группу является успешное завершение курса "Алгоритмы и структуры данных. С++. Уровень 2".

Форма обучения – очная.

Формирование групп целесообразно осуществлять с небольшим разбросом возраста и со схожим уровнем навыков решения олимпиадных задач по программированию. Оценка указанного уровня возможна на основании рекомендаций педагогов, ведущих программирование.

1.8. Формы организации занятий

Основной формой организации деятельности детей на занятии является групповое занятие. При изучении теоретических занятий педагог объясняет учебный материал с использованием слайдов и магнитно-маркерной доски.

Закрепление изученного материала осуществляется самостоятельной работой с помощью педагога по решению задач на компьютере с использованием специализированных сайтов.

Периодически проводятся олимпиады, предполагающие самостоятельную работу по решению задач на компьютере без помощи педагога.

После олимпиад, как правило, педагогом осуществляется разбор задач. Для разбора могут привлекаться обучаемые, успешно решившие те или иные задачи в ходе олимпиады.

2 Календарный учебный график

Режим организации занятий по данной дополнительной общеобразовательной программе определяется календарным учебным графиком и соответствует нормам, утвержденным СП 2.4.3648-20 "Санитарно-эпидемиологические требования к организациям воспитания и обучения, отдыха и оздоровления детей и молодежи", СанПиН 1.2.3685-21 "Гигиенические нормативы и требования к обеспечению безопасности и (или) безвредности для человека факторов среды обитания").

Начало занятий – 1 сентября.

Окончание занятий – 31 мая.

Всего учебных недель (продолжительность учебного года) – 37 недель

Объем учебных часов – 148 академических часов.

Режим обучения - 4 раза в неделю в зависимости от расписания.

Продолжительность академического часа - 50 минут

3 Учебный план

№ п/п	Название раздела, темы	Количество часов			Формы аттестации, контроля
		Всего	Теория	Практика	
1.	Введение	7	3	4	Текущий контроль
2.	Циклы	7	1	6	Текущий контроль
3.	Базовые задачи на реализацию	6	1	5	Текущий контроль
4.	Сортировки	7	1	6	Текущий контроль
5.	Виды поиска	7	1	6	Текущий контроль
6.	Стандартные структуры данных	6	1	5	Текущий контроль

7.	Знакомство с техниками программирования	7	1	6	Текущий контроль
8.	Динамическое программирование	7	1	6	Текущий контроль
9.	Графы. Основные понятия. DFS	8	1	7	Текущий контроль
10.	Поиск в ширину (BFS)	6	1	5	Текущий контроль
11	Алгоритм Дейкстры	6	1	5	Текущий контроль
12	Геометрия	8	1	7	Текущий контроль
13	Алгоритм Флойда — Уоршелла. Поиск кратчайших путей между всеми парами вершин	6	1	5	Текущий контроль
14	Алгоритм Форда-Беллмана. Поиск кратчайших путей во взвешенном графе	6	1	5	Текущий контроль
15	Основные деревья. Алгоритмы Прима и Краскала. Система непересекающихся множеств. Дерево Фенвика.	6	1	5	Текущий контроль
16	Корневая эвристика. SQRТ-декомпозиция. Алгоритм Мо	6	1	5	Текущий контроль
17	Строковые алгоритмы	6	1	5	Текущий контроль
18	Дерево отрезков	6	1	5	Текущий контроль
19	Бор	6	1	5	Текущий контроль
20	Хеширование. Алгоритм Рабина-Карпа	6	1	5	Текущий контроль
21	ДП по маскам. Гамильтоновы графы и задача Коммивояжера	6	1	5	Текущий контроль

22	Декартово дерево	6	1	5	Текущий контроль
23	Декомпозиции деревьев	6	1	5	Текущий контроль
	Итого	148	25	123	

4 Содержание программы

Тема Введение

Цели и задачи:

Сформировать у учащихся базовые навыки программирования на C++, необходимые для дальнейшего изучения алгоритмов и структур данных. Обеспечить уверенное владение фундаментальными конструкциями языка. Изучить основные арифметические операции, приоритет операций и типы данных для вычислений. Познакомить с методами оптимизации и написания эффективных арифметических выражений. Сформировать понимание логических операций и условных конструкций. Ввести понятие тернарного оператора как компактной альтернативы условному оператору и отработать его применение на практике.

Результаты освоения:

Учащийся будет знать: синтаксис и принципы работы операторов ввода и вывода, основные арифметические и логические операторы языка C++, синтаксис условных конструкций и тернарного оператора, особенности работы с целочисленными и вещественными типами данных. Учащийся будет уметь: Реализовывать консольные программы, взаимодействующие с пользователем. Выполнять арифметические вычисления, правильно используя типы данных и учитывая приоритет операций. Применять условные операторы для реализации разветвляющихся алгоритмов. Использовать тернарный оператор для написания лаконичного кода в простых ситуациях выбора.

Краткое содержание занятий:

Ввод-вывод. Арифметика. Знакомство со средой разработки. Структура простой программы на C++. Стандартные потоки ввода и вывода. Хитрая арифметика. Условные выражения. Тернарный оператор. Логические операции сравнения и логические операторы. Конструкция ветвления.

Тема Циклы

Цели и задачи:

Сформировать у учащихся фундаментальное понимание концепции циклов как базового инструмента управления порядком выполнения программы. Научить применять циклы для автоматизации повторяющихся действий и обработки последовательностей данных. Развить алгоритмическое

мышление через решение практических задач. Освоить применение циклов для решения задач, не требующих использования массивов. Сформировать навык работы с индексами для последовательного обхода элементов массива.

Результаты освоения:

Учащийся будет знать: Синтаксис и семантику циклов. Понятия инициализации, условия остановки и модификации счетчика цикла. Базовые алгоритмы обработки данных в массивах. Учащийся будет уметь: Выбирать оптимальный тип цикла для решения конкретной задачи. Реализовывать алгоритмы с использованием вложенных циклов. Корректно использовать индексацию для работы с элементами массивов внутри циклов. Реализовывать типовые операции над массивами: поиск, суммирование, проверка условий. Отлаживать программы, содержащие циклы, и избегать типичных ошибок.

Краткое содержание занятий:

Циклы без массивов. Понятие итерации. Управление циклами. Циклы и массивы. Модификация массивов. Вложенные циклы для обхода строк и столбцов.

Тема Базовые задачи на реализацию

Цели и задачи:

Закрепить на практике понимание фундаментальных конструкций языка C++ и базовых структур данных. Научить учащихся применять изученные концепции в комплексе для решения типовых алгоритмических задач. Научить преобразовывать словесное описание задачи в формальный алгоритм и рабочий код на C++. Закрепить использование циклов для обработки числовых последовательностей и строк. Сформировать умение работать с данными через основные структуры: числа, строки, массивы.

Результаты освоения:

Учащийся будет знать: Типовые формулировки и шаблоны решений для базовых алгоритмических задач. Понятие краевых случаев и их важность для корректности программы. Учащийся будет уметь: Анализировать условие задачи, выделять ключевые требования и входные/выходные данные. Составлять план решения (алгоритм) перед написанием кода. Проверять свое решение на различных наборах входных данных, включая краевые случаи.

Краткое содержание занятий:

Введение в решение задач. Задачи на числа и циклы. Задачи на обработку строк. Задачи на одномерные массивы. Комбинированные задачи.

Тема Сортировки

Цели и задачи:

Сформировать у учащихся системное понимание ключевых алгоритмов сортировки, их внутреннего устройства и областей применения. Научить

анализировать эффективность алгоритмов через оценку временной и пространственной сложности. Развить навык практической реализации основных сортировок на C++ и применения современных инструментов языка.

Результаты освоения:

Учащийся будет знать: Определения, алгоритмы и ключевые идеи основных квадратичных сортировок, быстрой сортировки, сортировки слиянием и сортировки подсчётом. Понятия устойчивости и неустойчивости сортировки. Принцип работы компаратора и правила его написания в C++. Учащийся будет уметь: Самостоятельно реализовывать изученные алгоритмы сортировки на C++. Анализировать эффективность алгоритма. Сравнить производительность алгоритмов на практических примерах.

Краткое содержание занятий:

Квадратичные сортировки. Сортировка слиянием. Быстрая сортировка. Сортировка с компаратором. Структуры данных. Методы подсчёта.

Тема Виды поиска

Цели и задачи:

Сформировать у учащихся системное понимание основных алгоритмов поиска, их принципов работы, сильных и слабых сторон. Научить выбирать оптимальный алгоритм поиска в зависимости от условий задачи. Закрепить навыки анализа временной и пространственной сложности алгоритмов. Практически освоить реализацию линейного, бинарного и тернарного поиска на языке C++ для работы как с простыми массивами, так и с пользовательскими структурами данных.

Результаты освоения:

Учащийся будет знать: Определения, алгоритмы и области применения линейного, бинарного и тернарного поиска. Необходимые предпосылки для применения бинарного и тернарного поиска (отсортированность данных). Учащийся будет уметь: Корректно реализовывать алгоритмы линейного, бинарного и тернарного поиска на C++. Анализировать входные данные и выбирать наиболее эффективный алгоритм.

Краткое содержание занятий:

Линейный поиск. Понятие линейного поиска. Алгоритм полного перебора. Бинарный поиск. Классическая реализация бинарного поиска. Тернарный поиск. Обобщение бинарного поиска. Реализация тернарного поиска.

Тема Стандартные структуры данных

Цели и задачи:

Сформировать у учащихся системное представление о фундаментальных абстрактных типах данных и их физических реализациях в C++. Развить умение

выбирать оптимальную структуру данных в зависимости от решаемой задачи. Закрепить понимание принципов их работы, сильных и слабых сторон через практическое применение.

Результаты освоения:

Учащийся будет знать: определение стека, основные операции и их сложность, области применения стека, разницу между стеком и очередью. Учащийся будет уметь: понимать ограничения реализации на основе массива.

Краткое содержание занятий:

Введение в стек. Очередь. Приоритетная очередь. Множества и словари. Операции поиска, вставки, удаления.

Тема Знакомство с техниками программирования

Цели и задачи:

Сформировать у учащихся базовое понимание методов решения алгоритмических задач. Развить навык анализа задачи и выбора подходящего метода решения. Познакомить с концепцией эффективности алгоритма (сложность по времени и памяти) на практических примерах. Научить реализовывать основные техники перебора и жадные алгоритмы на языке C++. Заложить фундамент для дальнейшего изучения более сложных алгоритмов и структур данных.

Результаты освоения:

Учащийся будет знать: Определение и признаки жадных алгоритмов, их сильные и слабые стороны. Понятия полного перебора, рекурсии и перебора подмножеств с помощью битовых масок. Учащийся будет уметь: Анализировать условие задачи и определять применимость жадного подхода. Реализовывать линейные и вложенные циклы для решения задач методом базового перебора. Сравнить эффективность разных подходов к решению одной задачи.

Краткое содержание занятий:

Жадные алгоритмы и базовые техники. Базовый перебор. Рекурсивный перебор. Введение в метод. Перебор масок. Использование битовых операций в C++ для представления подмножеств.

Тема Динамическое программирование

Цели и задачи:

Сформировать понимание парадигмы динамического программирования как метода решения оптимизационных задач путём разбиения на перекрывающиеся подзадачи. Изучить отличие ДП от полного перебора и жадных алгоритмов. Научиться выявлять подзадачи и строить таблицу (массив) состояний. Научиться применять ДП для решения классических задач.

Результаты освоения:

Знать основные понятия ДП (состояние, переход, база), виды подходов. Уметь формализовать задачу для решения методом ДП: определять состояние, базовые случаи и рекуррентные соотношения. Анализировать сложность решения.

Краткое содержание занятий:

Основы. Классические задачи. Задачи на последовательности и подобное. Игры. Новые задачи. Простые игры. Теория Шпрага-Гранди. Сложные задачи.

Тема Графы. Основные понятия. DFS**Цели и задачи:**

Сформировать у учащихся фундаментальное понимание структуры данных «граф» и её ключевых элементов. Освоить применение DFS для решения классического набора задач: проверка графа на связность, нахождение компонент связности, проверка на двудольность, поиск циклов, топологическая сортировка. Научить учащихся анализировать сложность алгоритма DFS.

Результаты освоения:

Учащийся будет знать: Определение графа, его основные элементы и их свойства. Основные виды графов. Принцип работы алгоритма поиска в глубину (DFS). Учащийся будет уметь: Считывать и хранить граф в программе, используя наиболее подходящее представление. Применять модификации DFS для решения задач.

Краткое содержание занятий:

Введение в теорию графов. Основные понятия и представления. Базовые задачи. Простой обход. Компоненты связности. Двудольность. Удалённость вершины. Поиск цикла. Игры на графах. TopSort, КСС и конденсация. Мосты и точки сочленения. Эйлеровы графы. Перебор. Продвинутые задачи. ДП на дереве. Переливайка. Наименьший общий предок. LCA, LA.

Тема Поиск в ширину (BFS)**Цели и задачи:**

Сформировать у учащихся понимание алгоритма поиска в ширину (BFS) как фундаментального инструмента для обхода графов. Изучить принцип работы алгоритма, основанный на постепенном исследовании вершин по уровням удаленности от стартовой точки. Освоить практическую реализацию BFS на C++. Научить применять алгоритм для решения классических задач на графах: поиск кратчайшего пути в невзвешенном графе, проверка графа на двудольность, подсчет компонент связности.

Результаты освоения:

Учащийся будет знать: Определение и принцип работы алгоритма BFS. Основные области применения BFS: поиск кратчайшего пути, анализ свойств графа. Учащийся будет уметь: Реализовывать алгоритм BFS на C++ для графов.

Краткое содержание занятий:

Теоретические основы BFS. Понятие графа: вершины, рёбра, представление графа. Идея алгоритма. Базовые задачи. Продвинутые задачи.

Тема Алгоритм Дейкстры

Цели и задачи:

Сформировать у учащихся глубокое понимание алгоритма Дейкстры как фундаментального метода. Развить навыки анализа графовых задач и выбора оптимального алгоритма для их решения. Освоить базовую и оптимизированную реализации алгоритма на C++. Научить применять алгоритм Дейкстры для решения классических и олимпиадных задач.

Результаты освоения:

Учащийся будет знать: Принцип работы алгоритма Дейкстры, его шаги и инварианты. Классические задачи, решаемые с помощью данного алгоритма. Учащийся будет уметь: Выбирать алгоритм Дейкстры как подходящее решение для задачи на кратчайшие пути. Проводить анализ временной сложности своего решения.

Краткое содержание занятий:

Реализация. Классические задачи. Трудная реализация. Нюансы и оптимизации.

Тема Геометрия

Цели и задачи:

Сформировать у учащихся базовое понимание о применении вычислительной геометрии в программировании и развить навыки решения классических задач.

Результаты освоения:

Учащийся будет знать: Основные геометрические понятия и их представление в программе. Признаки и условия, используемые в базовых геометрических задачах. Учащийся будет уметь: Анализировать условие задачи и сводить его к комбинации изученных базовых алгоритмов.

Краткое содержание занятий:

Введение в вычислительную геометрию. Точки, векторы, базовые операции. Задачи на взаимное расположение объектов. Вычисление площадей.

Практикум по решению задач СПбГУ. Простые задачи. Задачи на окружности. Многоугольники и выпуклые оболочки. Геометрические преобразования.

Тема Алгоритм Флойда — Уоршелла. Поиск кратчайших путей между всеми парами вершин

Цели и задачи:

Изучить принцип работы алгоритма Флойда-Уоршелла. Научиться анализировать область применения алгоритма. Практически реализовать алгоритм на C++ и применять его для решения типовых задач.

Результаты освоения:

Учащийся будет знать: Постановку задачи поиска кратчайших путей между всеми парами вершин. Основные сферы применения алгоритма. Учащийся будет уметь: Реализовывать алгоритм Флойда-Уоршелла на C++.

Краткое содержание занятий:

Введение в проблему всех пар вершин. Динамическое программирование как основа алгоритма. Практическая реализация на C++. Анализ и применение.

Тема Алгоритм Форда-Беллмана. Поиск кратчайших путей во взвешенном графе

Цели и задачи:

Сформировать понимание задачи поиска кратчайших путей в графах. Изучить принципы и динамическую логику, лежащую в основе алгоритма Беллмана-Форда. Закрепить материал через решение практических задач, моделирующих реальные ситуации.

Результаты освоения:

Учащийся будет знать: Постановку задачи поиска кратчайших путей в графах. Алгоритмические шаги Беллмана-Форда и его псевдокод. Учащийся будет уметь: Реализовывать алгоритм Беллмана-Форда на C++ с использованием удобных структур данных. Применять алгоритм для решения типовых задач.

Краткое содержание занятий:

Введение в проблему. Принцип релаксации. Динамическое программирование как основа алгоритма. Алгоритмические шаги Беллмана-Форда.

Тема Остовные деревья. Алгоритмы Прима и Краскала. Система непересекающихся множеств. Дерево Фенвика.

Цели и задачи:

Сформировать понимание теории графов на примере задачи поиска минимального остовного дерева. Изучить концепции взвешенного графа, остовного дерева и его минимальности. Изучить структуру данных «Дерево Фенвика». Рассмотреть классические и продвинутые задачи, решаемые с помощью этих алгоритмов и структур данных.

Результаты освоения:

Учащийся будет знать: Определения: взвешенный граф, остовное дерево, минимальное остовное дерево. Принципы «жадных» алгоритмов, их шаги и отличия. Принцип работы дерева Фенвика. Учащийся будет уметь: Применять изученные структуры для решения задач остовного дерева и не только.

Краткое содержание занятий:

Введение в остовные деревья. Алгоритм Краскала. Алгоритм Прима. Сравнение алгоритмов. Дерево Фенвика. Базовые задачи.

Тема Корневая эвристика. SQRT-декомпозиция. Алгоритм Мо.**Цели и задачи:**

Сформировать понимание философии корневой эвристики (SQRT-оптимизации) как мощного метода борьбы с неэффективностью алгоритмов путём разбиения данных на блоки. Изучить структуру данных SQRT-декомпозиция, её принципы построения и выполнения запросов на отрезке. Освоить техники анализа сложности операций в SQRT-декомпозиции. Реализовать на C++ основные паттерны SQRT-декомпозиции и алгоритма Мо.

Результаты освоения:

Учащийся будет знать: Определение и идею корневой эвристики. Принцип построения структуры данных SQRT-декомпозиция. Суть алгоритма Мо: декомпозицию запросов, сортировку по блокам. Области применения и ограничения данных методов. Учащийся будет уметь: Определять, применима ли корневая эвристика или алгоритм Мо к задаче. Проводить сравнительный анализ эффективности с другими структурами данных.

Краткое содержание занятий:

Введение в корневую эвристику. SQRT-декомпозиция на практике. Запросы на отрезке. Алгоритм Мо. Декомпозиция запросов. Разделение объектов. Практикум по алгоритму Мо.

Тема Строковые алгоритмы**Цели и задачи:**

Сформировать у учащихся понимание необходимости эффективной обработки строковых данных и познакомить их с ключевыми алгоритмами,

лежащими в основе современной текстовой обработки. Освоить алгоритм КМП для эффективного поиска подстроки, избавившись от наивного подхода. Освоить вычисление Z-функции и её применение для решения задач, альтернативных префикс-функции. Изучить алгоритм Манакера для линейного поиска всех палиндромов в строке. Научить применять изученные алгоритмы для решения олимпиадных и практических задач.

Результаты освоения:

Учащийся будет знать: Определения и принципы работы префикс-функции, Z-функции. Устройство и логику алгоритмов КМП и Манакера. Учащийся будет уметь: Реализовывать на C++ вычисление префикс-функции и Z-функции. Выбирать оптимальный строковый алгоритм под конкретную задачу.

Краткое содержание занятий:

Введение. Хитрая задача. Префикс-функция. Алгоритм КМП. Z-функция. Алгоритм Манакера.

Тема Дерево отрезков

Цели и задачи:

Углубить знания учащихся о структурах данных для обработки запросов на массивах. Сформировать полное понимание рекурсивной структуры и идеологии «дерева отрезков» (ДО). Научить реализовывать базовые и модифицированные версии ДО на C++ для эффективного решения задач на последовательностях. Развить навыки анализа сложности и выбора оптимальной структуры данных под конкретную задачу.

Результаты освоения:

Учащийся будет знать: Принципы построения дерева отрезков, его структуру в памяти. Основные операции ДО. Области применения и ограничения дерева отрезков. Учащийся будет уметь: Реализовывать на C++ дерево отрезков для различных операций.

Краткое содержание занятий:

Базовые задачи. Sparse Table. Сравнение Sparse Table и ДО. Классические задачи. Scanline. Решение задач на плоскости методом сканирующей прямой. Нестандартные варианты ДО.

Тема Бор

Цели и задачи:

Сформировать понимание древовидной структуры данных «Бор», её принципов, преимуществ и ограничений. Изучить внутреннее устройство бора

и его ключевые отличия от других структур данных. Освоить основные операции работы с бором. Реализовать бор на C++.

Результаты освоения:

Учащийся будет знать: Определение бора, его внутреннее устройство. Область применения бора. Учащийся будет уметь: Реализовывать структуру узла и класса бора на C++. Корректно выполнять основные операции: вставка, удаление, поиск. Применять бор для решения практических задач, связанных со строками.

Краткое содержание занятий:

Теоретические основы бора. Реализация бора на C++. Основные операции: поиск и удаление. Решение прикладных задач.

Тема Хеширование. Алгоритм Рабина-Карпа

Цели и задачи:

Сформировать глубокое понимание концепции хеширования и его применения для эффективного сравнения данных. Изучить принципы работы алгоритма Рабина-Карпа, его сильные и слабые стороны. Реализовать алгоритм Рабина-Карпа на C++ для решения задачи поиска подстроки и других связанных задач. Рассмотреть классические и олимпиадные задачи, решаемые с помощью хеширования.

Результаты освоения:

Учащийся будет знать: Определение и области применения хеширования. Принцип работы и шаги алгоритма Рабина-Карпа. Учащийся будет уметь: Реализовывать алгоритм Рабина-Карпа на C++.

Краткое содержание занятий:

Теоретические основы хеширования. Алгоритм Рабина-Карпа: принцип работы и реализация. Базовые задачи. Стандартные задачи. Сложные задачи.

Тема ДП по маскам. Гамильтоновы графы и задача Коммивояжёра

Цели и задачи:

Сформировать глубокое понимание концепции динамического программирования по подмножествам (ДП по маскам) как мощного инструмента для решения задач. Изучить теорию гамильтоновых графов. Освоить постановку и решение классической задачи Коммивояжёра.

Результаты освоения:

Учащийся будет знать: Определение гамильтонова цикла и пути. Суть задачи Коммивояжёра и её практическую значимость. Стандартную формулу ДП для решения TSP. Ограничения метода ДП по маскам. Учащийся будет

уметь: Реализовывать решение задачи Коммивояжёра и других подобных задач на языке C++.

Краткое содержание занятий:

Введение в ДП по подмножествам. Понятие гамильтонова цикла. Задача Коммивояжёра как классический пример трудной задачи. Решение задачи Коммивояжёра методом ДП. Простые задачи. Усложненные задачи. ДП по профилю.

Тема Декартово дерево

Цели и задачи:

Изучить принцип работы декартова дерева. Освоить основные операции: разделение и слияние как фундаментальные для работы со структурой. Реализовать на C++ декартово дерево как по явному, так и по неявному ключу. Сформировать навыки анализа времени выполнения операций.

Результаты освоения:

Учащийся будет знать: Определение декартова дерева, его основные свойства (приоритет и ключ). Понятия явного и неявного ключа, их различия и области применения. Учащийся будет уметь: Реализовывать на C++ структуру узла и основные операции декартова дерева. Решать сложные задачи, требующие модификации и агрегации данных в дереве.

Краткое содержание занятий:

Введение в Декартово дерево. Базовые операции. Базовые задачи. Неявный ключ. Преобразование декартова дерева в структуру для работы с последовательностями. Сложные задачи. Разбор сложных олимпиадных задач.

Тема Декомпозиции деревьев

Цели и задачи:

Сформировать у учащихся системное понимание двух фундаментальных методов декомпозиции деревьев — Heavy-Light и центроидной. Развить навыки анализа структуры дерева и применения её свойств для решения сложных задач. Научить выбирать оптимальный метод декомпозиции в зависимости от постановки задачи. Сформировать практические навыки реализации этих методов на C++.

Результаты освоения:

Учащийся будет знать: Определения, принципы построения и свойства Heavy-Light и центроидной декомпозиций. Типовые задачи, решаемые с помощью каждой из декомпозиций. Учащийся будет уметь: Реализовывать алгоритмы построения HLD и центроидной декомпозиции на C++.

Краткое содержание занятий:

Введение в декомпозиции деревьев. Heavy-Light декомпозиция (HLD). Практическое применение HLD. Центроидная декомпозиция. Практическое применение центроидной декомпозиции. Сравнительный анализ и решение комплексных задач.

5 Оценочные и методические материалы

5.1. Формы аттестации

Для оценки результативности обучения по дополнительной общеобразовательной программе «Алгоритмы и структуры данных. C++. Уровень 3» применяется входной, текущий, промежуточный контроль.

Входной контроль

В качестве входного контроля - см. пункт **Условия реализации программы**

Текущий контроль

Осуществляется автоматизировано с помощью автоматизированной системы.

Раздел считается освоенным, если задачи решены в объеме не менее 80% от числа задач, входящих в набор. Если задачи наборов решались с существенной помощью педагога, обучаемому предлагается перерешать задачи наборов.

Фиксация результатов осуществляется автоматически.

1 Промежуточный контроль

Проводится в форме олимпиады по программированию.

Итоговый контроль

Для определения итогов освоения программы используются результаты промежуточного контроля.

5.2. Методическое обеспечение

Дидактический материал

№ п/п	Тема	Формы занятий	Дидактический материал
1.	Введение	Лекция, Практика	Лекция.
2.	Циклы	Лекция, Практика	Лекция. Слайды
3.	Базовые задачи на реализацию	Лекция, Практика	Лекция.
4.	Сортировки	Лекция, Практика	Лекция.

5.	Виды поиска	Лекция, Практика	Лекция. Слайды
6.	Стандартные структуры данных	Лекция, Практика	Лекция. Слайды
7.	Знакомство с техниками программирования	Лекция, Практика	Лекция. Слайды
8.	Динамическое программирование	Лекция, Практика	Слайды
9.	Графы. Основные понятия. DFS	Лекция, Практика	Лекция. Слайды
10.	Поиск в ширину (BFS)	Лекция, Практика	Лекция. Слайды
11.	Алгоритм Дейкстры	Лекция, Практика	Лекция. Слайды
12.	Геометрия	Лекция, Практика	Лекция. Слайды
13.	Алгоритм Флойда — Уоршелла. Поиск кратчайших путей между всеми парами вершин	Лекция, Практика	Лекция. Слайды
14.	Алгоритм Форда-Беллмана. Поиск кратчайших путей во взвешенном графе	Лекция, Практика	Лекция. Слайды
15.	Остовные деревья. Алгоритмы Прима и Краскала. Система непересекающихся множеств. Дерево Фенвика.	Лекция, Практика	Лекция. Слайды
16.	Корневая эвристика. SQRT-декомпозиция. Алгоритм Мо	Лекция, Практика	Лекция. Слайды
17.	Строковые алгоритмы	Лекция, Практика	Лекция. Слайды
18.	Дерево отрезков	Лекция, Практика	Лекция. Слайды
19.	Бор	Лекция, Практика	Лекция. Слайды
20.	Хеширование. Алгоритм Рабина-Карпа	Лекция, Практика	Лекция. Слайды
21.	ДП по маскам. Гамильтоновы графы и задача Коммивояжёра	Лекция, Практика	Лекция. Слайды
22.	Декартово дерево	Лекция, Практика	Лекция. Слайды
23.	Декомпозиции деревьев	Лекция, Практика	Лекция. Слайды

6 Материально-техническое обеспечение программы

6.1. Перечень необходимого оборудования

Техническое оснащение:

- Кабинет, имеющий хорошее освещение.
- Компьютеры (ноутбуки) из расчета один на обучаемого, с доступом к сети Интернет.
- Стулья, рабочие столы.
- Стол и стул педагога.
- Доска магнитно-маркерная.
- Проектор

Программные средства:

- Visual Studio
- Code::Blocks

6.2. Учебно-методическое обеспечение:

1. Слайды по изучаемым темам.
2. Сайты с подготовленными разделами для реализации данной программы.

7 Кадровое обеспечение

Согласно Профессиональному стандарту «Педагог дополнительного образования детей и взрослых» по данной программе может работать педагог дополнительного образования с уровнем образования и квалификации, соответствующим обозначениям таблицы пункта 2 Профессионального стандарта (Описание трудовых функций, входящих в профессиональный стандарт), а именно: коды А и В с уровнями квалификации 6.

8 Список литературы

8.1. Литература для педагога

1. Федеральный закон «Об образовании в Российской Федерации» №273-ФЗ от 29.12.2012 (ст.2, п.9, п. 14; ст.12, п.5; ст.33, п.2; ст.75, п.2, п.4)
2. Приказ Министерства просвещения РФ от 9 ноября 2018 г. N 196 "Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам"
3. СП 2.4.3648-20 "Санитарно-эпидемиологические требования к организациям воспитания и обучения, отдыха и оздоровления детей и молодежи"
4. СанПиН 1.2.3685-21 "Гигиенические нормативы и требования к обеспечению безопасности и (или) безвредности для человека факторов среды обитания"
5. Концепция развития дополнительного образования детей от 04.09.2014 № 1726-р (раздел IV)

6. Методические рекомендации по проектированию дополнительных общеразвивающих программ (Приложение к письму Департамента государственной политики в сфере воспитания детей и молодежи Министерства образования и науки РФ от 18.11.2015 № 09-3242)
7. А. В. Андреева, А. С. Станкевич. «Программирование: теоремы и задачи».
8. А. Хирвонен, Т. Хирвонен. «Справочник по олимпиадному программированию на C++».
9. Олимпиадное программирование / Антти Лааксонен; пер. с англ. А.А. Слинкин – М.:ДМК Пресс, 2018. -300 с.: ил.

8.2. Литература для учащихся

10. С. А. Алексеев, О. В. Гасников и др. «Олимпиадное программирование в ЛКШ. Зеленый и красный трактаты».

8.3. Интернет-ресурсы

11. codeforces.com
12. informatics.msk.ru
13. lksh.ru
14. sochisirius.ru