

37. АЛГОРИТМЫ И СТРУКТУРЫ ДАННЫХ В ДИСКРЕТНОЙ МАТЕМАТИКЕ И ИХ ПРИЛОЖЕНИЯ

Касьянова А.И., студент гр.373904, Русина Н. В., аспирант

*Белорусский государственный университет информатики и радиоэлектроники'
г. Минск, Республика Беларусь*

Ефремов А.А. – канд. экон. наук, доцент каф. ЭИ

Аннотация. Данная научная работа сосредоточена на изучении алгоритмов и структур данных в контексте дискретной математики. В работе проводится анализ различных алгоритмических подходов и структур данных, используемых для решения задач, основанных на дискретной математике.

Ключевые слова. Алгоритм, поиск, сортировка, графы, структура данных, матрица, деревья.

Поиск и сортировка – это ключевые концепции в области алгоритмов и структур данных, которые имеют важную роль в программировании и информационных технологиях. Рассмотрим эти концепции более подробно.

Поиск: алгоритмы поиска применяются для нахождения конкретного элемента в наборе данных. Существуют два основных алгоритма поиска – линейный и бинарный.

Линейный поиск: простой алгоритм, который перебирает элементы последовательности по очереди и сравнивает их с искомым значением, пока не будет найден искомый элемент или не будут

просмотрены все элементы. В связи с малой эффективностью по сравнению с другими алгоритмами линейный поиск обычно используют, только если отрезок поиска содержит очень мало элементов [1].

Бинарный поиск: эффективный алгоритм для поиска в отсортированных массивах. Он сравнивает искомое значение с элементом в середине массива и определяет, в какой половине массива продолжать поиск, уменьшая область поиска вдвое на каждом шаге. [2]

Рассмотрим более подробно алгоритмы сортировки.

Алгоритмы сортировки используются для упорядочивания элементов в наборе данных по возрастанию или убыванию. Имеется множество алгоритмов сортировки, однако некоторые из наиболее известных включают:

Сортировка пузырьком: простой алгоритм, который не единожды проходит по списку и меняет местами соседние элементы, если они находятся не в нужном порядке. В сравнении с иными алгоритмами считается элементарным для понимания и осуществления. Эффективен для массивов небольшого размера.

Сортировка вставками: алгоритм, который строит отсортированную последовательность по одному элементу за раз. Эффективен для небольших списков.

Быстрая сортировка: один из наиболее действенных алгоритмов сортировки, он использует метод «разделяй и властвуй». Быстрая сортировка разбивает список на меньшие подсписки, сортирует их отдельно и затем объединяет весь список. Первоначально производятся перестановки на наибольшем возможном расстоянии и после каждого прохода элементы делятся на две независимые группы.[3]

Так же существуют графовые алгоритмы, рассмотрим алгоритмы обхода графов.

Графовые алгоритмы – это ценная часть теории графов, которая отвечает за разработку методов обработки и анализа графовых структур. Изучим некоторые из основных графовых алгоритмов:

Обход графов:

– DFS (Depth–First Search) – это алгоритм поиска в глубину. Он изучает граф, спускаясь «внутрь» структуры до момента, пока не будет достигнут последний узел или не будут просмотрены все узлы.

– BFS (Breadth–First Search) – это алгоритм поиска в ширину. Он исследует граф, движется от первоначального узла ко всем его соседям, затем ко всем соседям соседей и так далее.

Поиск кратчайшего пути:

– алгоритм Дейкстры – используется для нахождения кратчайшего пути от одного из узлов графа ко всем остальным. Он работает только с неотрицательными весами ребер.

– алгоритм Флойда–Уоршелла – находит кратчайшие пути между всеми парами вершин во взвешенном графе. Он способен работать с отрицательными весами ребер, но не с отрицательными циклами.

Поиск минимального остовного дерева:

– алгоритм Прима – находит минимальное остовное дерево во взвешенном связанном графе. Начинает работу с одной вершины и постепенно добавляет ребра к дереву, выбирая ребро минимального веса.

– алгоритм Крускала – также находит минимальное остовное дерево. Но сначала сортирует все ребра по весу и затем постепенно добавляет ребра к остову, избегая образования циклов.

Эти алгоритмы играют важную роль в решении различных задач, связанных с графами, таких как поиск кратчайших путей, построение сетей связности, анализ социальных сетей и многих других приложений. Они составляют основу для многих графовых алгоритмов и имеют широкое применение в информатике и программировании.

Далее проанализируем структуры данных.

В дискретной математике массивы и связанные списки также имеют особую роль, особенно в контексте структур данных и алгоритмов. Рассмотрим их специфику и возможные операции в этом контексте:

Массивы:

– специфика: в дискретной математике массив демонстрирует собой упорядоченную коллекцию элементов, индексируемую целыми числами.

– операции: доступ к элементу по индексу: массивы обеспечивают быстрый доступ к элементу по его индексу, что значительно для многих алгоритмов и операций в дискретной математике.

– математические операции: массивы могут представлять матрицы, вектора и другие математические структуры, что может помочь выполнить задачи линейной алгебры и других математических операций.

Связанные списки:

– специфика: в дискретной математике связанные списки могут быть полезны для представления структур данных, например для списков смежности для графов или для реализации других алгоритмов.

– операции: вставка и удаление: связанные списки могут действительно поддерживать операции вставки и удаления, что так же помогает для написания многих алгоритмов обработки данных в дискретной математике.

– представление структур данных: связанные списки могут быть использованы для наглядности различных структур данных, играющих особую роль в дискретной математике, таких как списки, стеки, очереди и другие.

Так же рассмотрим одну из самых используемых и распространённых структур данных – деревья.

Бинарные деревья поиска:

– особенности: бинарное дерево поиска – это структура данных, в которой каждый узел имеет не более двух потомков, и узлы упорядочены таким образом, что для каждого узла все значения в левом поддереве меньше, чем значение в узле, а все значения в правом поддереве больше.

– применение: бинарные деревья поиска широко распространены для реализации операций поиска, вставки и удаления элементов, что делает их особым и незаменимым инструментом для хранения и обработки упорядоченных данных.

Красно-черные деревья:

– особенности: красно-черное дерево – это тип бинарного дерева поиска, в котором каждый узел содержит дополнительную информацию о «цвете» (красный или черный), и которое поддерживает баланс при вставке и удалении элементов.

– применение: красно-черные деревья позволяют эффективно выполнять операции поиска, вставки и удаления, сохраняя сбалансированную структуру дерева. Используются красно-черные деревья в разных алгоритмах и структурах данных, к примеру таких как ассоциативные массивы и базы данных.

В дискретной математике деревья играют важную роль, потому что они используются для моделирования и анализа различных структур данных, алгоритмов и математических концепций.

Применение в дискретной математике:

– оптимизация алгоритмов: деревья обладают незаменимой ролью в оптимизации алгоритмов поиска, сортировки, обхода и других операций над данными.

– моделирование и анализ: деревья используются для моделирования и анализа разных математических и дискретных структур, таких как графы, орграфы, деревья решений и другие.

Таким образом, можно сделать вывод что в дискретной математике деревья играют особую роль в представлении, обработке и анализе структур данных, алгоритмов и математических концепций. Они используются для моделирования различных структур, оптимизации алгоритмов и решения различных задач, связанных с обработкой и анализом дискретных данных.[4]

Рассмотрим структуру данных – графы.

В дискретной математике графы используются для моделирования различных связей и отношений между объектами. Изучим структуры данных, используемые для представления графов и их применение в дискретной математике.

Матрицы смежности:

– особенности: Матрица смежности представляет собой двумерный массив, в нем значение в позиции (i, j) указывает на наличие ребра между вершинами i и j ;

– применение: матрицы смежности используют для представления связей между вершинами в графах. Обеспечивают быстрый доступ к информации о смежных вершинах и быстром поиске ребер.

Списки смежности:

– особенности: список смежности несет собой массив списков, в котором каждый список имеет вершины, смежные с определенной вершиной.

– применение: списки смежности применяются для эффективного хранения информации о смежных вершинах в графе. Дает компактно и аккуратно сохранить информацию о связях в графе и обеспечивают удобный доступ к смежным вершинам.

Применение графов в дискретной математике:

– моделирование и анализ: Графы используют для моделирования различных дискретных структур, таких как сети, связности, деревья, орграфы и другие. Использование различных структур данных для представления графов может помочь проводить анализ и исследование их свойств, а также решать различные задачи, связанные с графами.

Алгоритмы и операции: в дискретной математике графы используются для реализации и анализа разных алгоритмов, к примеру поиск путей, обходы графов, поиск связных компонент, поиск циклов и др.

Таким образом, в дискретной математике структуры данных для представления графов важны в моделировании, анализе и решении разного рода задач, которые связанные с дискретными структурами и алгоритмами. Использование матриц смежности и списков смежности позволяет наглядно представить

*60-я Юбилейная Научная Конференция Аспирантов, Магистрантов и Студентов БГУИР,
Минск 2024*

информацию о связях и отношениях в графах, это делает их важным инструментом для анализа и обработки дискретных данных.

Список использованных источников:

1. Электронный ресурс: *Линейный поиск* – Википедия (wikipedia.org)
2. Электронный ресурс: <https://swiftbook.org/tutorials/binarnyy-poisk/?ysclid=luv5oqy4qo143499897>.
3. Электронный ресурс: [Быстрая сортировка](https://wikipedia.org) — Википедия (wikipedia.org).
4. Электронный ресурс: https://www.bsuir.by/m/12_100229_1_90178.pdf?ysclid=luv5tyvvla501786739.
5. Электронный ресурс: https://portal.tpu.ru/SHARED/t/TRACEY/Courses/DisMath/Tab/DiscreteMathIDO_2016_part1.pdf.