Министерство Образования, Культуры и Исследований

Молдавский Государственный Университет

Факультет Математики и Информатики

Департамент Информатики

**Реферат**

Алгоритмы и структуры данных

На тему: “Методы и процедуры поиска”

**Выполнил:**

студент группы IA2102

Maxim Ianac

Кишинев 2023

Введение

В компьютерных науках алгоритм поиска — это алгоритм (обычно включающий множество других, более конкретных алгоритмов), решающий задачу поиска. Алгоритмы поиска работают для извлечения информации, хранящейся в некоторой структуре данных или рассчитанной в пространстве поиска проблемной области, с дискретными или непрерывными значениями.

Хотя задачи поиска, описанные выше, и веб-поиск являются проблемами информационного поиска, они обычно изучаются как отдельные области и решаются и оцениваются по-разному. Проблемы веб-поиска обычно сосредоточены на фильтрации и поиске документов, наиболее релевантных человеческим запросам. Классические алгоритмы поиска обычно оцениваются по тому, насколько быстро они могут найти решение и гарантируется ли это решение оптимальным. Хотя алгоритмы поиска информации должны быть быстрыми, важнее качество ранжирования, а также то, были ли исключены хорошие результаты и включены плохие.

Соответствующий алгоритм поиска часто зависит от искомой структуры данных и может также включать в себя предварительную информацию о данных. Алгоритмы поиска можно сделать быстрее или эффективнее за счет специально созданных структур базы данных, таких как деревья поиска, хэш-карты и индексы базы данных.

Алгоритмы поиска можно разделить на основе их механизма поиска на три типа алгоритмов: линейные, бинарные и хэширующие. Алгоритмы линейного поиска проверяют каждую запись на наличие записи, связанной с целевым ключом, линейным способом. Бинарный поиск, или полуинтервал, постоянно нацелен на центр структуры поиска и делит пространство поиска пополам. Алгоритмы сравнительного поиска улучшают линейный поиск, последовательно удаляя записи на основе сравнения ключей до тех пор, пока не будет найдена целевая запись, и могут применяться к структурам данных в определенном порядке. Алгоритмы цифрового поиска работают на основе свойств цифр в структурах данных, использующих цифровые ключи. Наконец, хеширование напрямую сопоставляет ключи с записями на основе хэш-функции.

Алгоритмы часто оцениваются по их вычислительной сложности или максимальному теоретическому времени выполнения. Например, функции двоичного поиска имеют максимальную сложность O(log n) или логарифмическое время. Это означает, что максимальное количество операций, необходимых для нахождения цели поиска, является логарифмической функцией размера области поиска.

Возможное применение методов поиска:

1. Проблемы комбинаторной оптимизации, такие как:
2. Проблемы с удовлетворением ограничений, такие как: Проблема раскраски карты Заполнение судоку или кроссворда
3. В теории игр и особенно в комбинаторной теории игр выбор лучшего следующего хода (например, в алгоритме минимакс)
4. Подбор комбинации или пароля из всего набора возможностей
5. Факторизация целого числа (важная проблема в криптографии)
6. Оптимизация промышленного процесса, такого как химическая реакция, путем изменения параметров процесса (таких как температура, давление и pH)
7. Получение записи из базы данных
8. Поиск максимального или минимального значения в списке или массиве
9. Проверка наличия заданного значения в наборе значений

Классы:

Для виртуальных поисковых пространств

Алгоритмы поиска виртуальных пространств используются в задаче удовлетворения ограничений, где цель состоит в том, чтобы найти множество присвоений значений определенным переменным, которые будут удовлетворять конкретным математическим уравнениям и неравенствам/равенствам. Они также используются, когда цель состоит в том, чтобы найти такое назначение переменной, которое максимизирует или минимизирует определенную функцию этих переменных. Алгоритмы для этих задач включают базовый поиск методом грубой силы (также называемый «наивным» или «неосведомленным» поиском) и различные эвристики, которые пытаются использовать частичное знание о структуре этого пространства, такие как линейная релаксация, генерация ограничений, и распространение ограничений.

Важным подклассом являются методы локального поиска, которые рассматривают элементы пространства поиска как вершины графа с ребрами, определяемыми набором эвристик, применимых к данному случаю; и сканировать пространство, перемещаясь от элемента к элементу по краям, например, по критерию наискорейшего спуска или лучшего первого, или в стохастическом поиске. Эта категория включает большое разнообразие общих метаэвристических методов, таких как имитация отжига, поиск табу, А-команды и генетическое программирование, которые определенным образом комбинируют произвольные эвристики. Противоположностью локальному поиску являются методы глобального поиска. Этот метод применим, когда пространство поиска не ограничено и все аспекты данной сети доступны объекту, выполняющему алгоритм поиска.

Этот класс также включает в себя различные алгоритмы поиска по дереву, которые рассматривают элементы как вершины дерева и обходят это дерево в определенном порядке. Примеры последнего включают исчерпывающие методы, такие как поиск в глубину и поиск в ширину, а также различные основанные на эвристике методы обрезки дерева поиска, такие как поиск с возвратом и ветвление и граница. В отличие от общей метаэвристики, которая в лучшем случае работает только в вероятностном смысле, многие из этих методов поиска по дереву гарантированно найдут точное или оптимальное решение, если им будет предоставлено достаточно времени. Это называется «полнота».

Другой важный подкласс состоит из алгоритмов исследования игрового дерева многопользовательских игр, таких как шахматы или нарды, узлы которого состоят из всех возможных игровых ситуаций, которые могут возникнуть в результате текущей ситуации. Цель в этих задачах — найти ход, обеспечивающий наилучшие шансы на победу, с учетом всех возможных ходов противника (соперников). Аналогичные проблемы возникают, когда люди или машины должны принимать последовательные решения, результаты которых не полностью находятся под их контролем, например, при управлении роботами или при планировании маркетинговой, финансовой или военной стратегии. Такого рода задачи — комбинаторный поиск — широко изучались в контексте искусственного интеллекта. Примерами алгоритмов для этого класса являются минимаксный алгоритм, альфа-бета-отсечение, а также алгоритм A\* и его варианты.

Для подструктур данной структуры

Название «комбинаторный поиск» обычно используется для алгоритмов, которые ищут определенную подструктуру данной дискретной структуры, такой как граф, строка, конечная группа и т. д. Термин комбинаторная оптимизация обычно используется, когда целью является поиск подструктуры с максимальным (или минимальным) значением некоторого параметра. (Поскольку подструктура обычно представляется в компьютере набором целочисленных переменных с ограничениями, эти задачи можно рассматривать как частные случаи удовлетворения ограничений или дискретной оптимизации, но обычно они формулируются и решаются в более абстрактной постановке, где внутреннее представление явно не упоминается.)

Важным и широко изученным подклассом являются алгоритмы графа, в частности алгоритмы обхода графа, для поиска конкретных подструктур в данном графе, таких как подграфы, пути, схемы и т. д. Примеры включают алгоритм Дейкстры, алгоритм Крускала, алгоритм ближайшего соседа и алгоритм Прима.

Другим важным подклассом этой категории являются алгоритмы поиска строк, которые ищут шаблоны в строках. Двумя известными примерами являются алгоритмы Бойера-Мура и Кнута-Морриса-Пратта, а также несколько алгоритмов, основанных на структуре данных суффиксного дерева.

Поиск максимума функции

В 1953 году американский статистик Джек Кифер разработал поиск Фибоначчи, который можно использовать для нахождения максимума унимодальной функции и который имеет множество других применений в информатике.

Для квантовых компьютеров

Существуют также методы поиска, разработанные для квантовых компьютеров, такие как алгоритм Гровера, которые теоретически быстрее, чем линейный поиск или поиск методом грубой силы, даже без помощи структур данных или эвристики. Хотя идеи и приложения, лежащие в основе квантовых компьютеров, все еще полностью теоретические, были проведены исследования с алгоритмами, подобными алгоритму Гровера, которые точно воспроизводят гипотетические физические версии систем квантовых вычислений.

Поисковая оптимизация:

Алгоритмы поиска, используемые в поисковой системе, такой как Google, упорядочивают релевантные результаты поиска на основе множества важных факторов. Поисковая оптимизация (SEO) — это процесс, в котором любой конкретный результат поиска будет работать в сочетании с поисковым алгоритмом, чтобы органично привлечь больше внимания, внимания и кликов на свой сайт. Это может зайти так далеко, как попытка настроить алгоритм поисковых систем, чтобы более активно использовать конкретный результат поиска, но стратегия, вращающаяся вокруг SEO, стала невероятно важной и актуальной в деловом мире.