

написана – C#, так как этот язык очень подходит для работы с классами, что для «плиточной» структуры является очень важным параметром.

Механизм игры крайне прост: имеется 48 заготовленных уровней. Все они разбиты на 4 категории по 12 уровней (4 на 4, 6 на 6, 8 на 8, 10 на 10). Изначально доступен первый уровень, после прохождения которого становится доступен второй. Методика открытия уровней повторяется до тех пор, пока не будет пройден последний 48 уровень. После этого готовые уровни заканчиваются и игроку становится доступен свободный режим: бесконечный уровень 10 на 10, в котором цвета плиток генерируются случайно.

Как же представлена запись готовых уровней? Имеется структура уровней, содержащая в себе идентификатор уровня (его номер), а также двумерный массив, отвечающий за вид формы. Каждый элемент массива может иметь значение 1 или 0. От этого числа зависит, какой стороной плитка будет повернута на рабочую камеру. Поэтому форма из плиток в коде представляется как квадратная матрица 2n-го порядка, в которой записаны единицы и нули.

У каждой плитки в Unity сцене есть свойство, которое отвечает за «рабочую» сторону. Также имеются свойства «Строка» и «Столбец». Благодаря этому за счет матрицы уровня мы можем задать нужной плитке определенный окрас, поворачивая или не поворачивая её.

Игра уже находится в свободном доступе на PlayMarket. Она бесплатна, но пока что доступна только для Android смартфонов. Требуемая версия Android - 2.3 и выше.

ЛИТЕРАТУРА

1. Приложения в Google Play / Plit: <https://play.google.com/store/apps/details?id=com.OrangeCompany.Plit>.

УДК 004.424.5

Студ. Е. А. Буйко, М. А. Скорина
Науч. рук. ст. преп. А. С. Наркевич
(кафедра информационных систем и технологий, БГТУ)

АНАЛИЗ АЛГОРИТМОВ СОРТИРОВОК

На сегодняшний день существует множество сортировок и каждая обладает своими особенностями. Цель данной работы заключается в объяснении разделения сортировок на различные виды, выделение основополагающих сортировок, разбор работы сортировок по памяти и времени, разбор применения сортировок с различными данными. С

каждым днём количество сортировок увеличивается, однако лишь на немногие из них стоит обратить пристальное внимание по причине их однообразности.

Разделение сортировок. Сортировки подразделяют на виды по различным свойствам: по алгоритму, по оптимальности, по сфере применения и т.д. В данной работе мы рассмотрим виды сортировок по алгоритму.

Обменная сортировка (включает в себя сортировки пузырьком, перемешиванием, быстрая, расчётской, сортировка чёт-нечёт) – это сортировка, которая выполняет обмен неупорядоченных элементов, пока такие существуют.

Сортировка сдвигом (включает в себя сортировки вставками, Шелла, деревом) – сортировка сдвигающая, согласно алгоритму, большие элементы в одну сторону, а меньшие в другую.

Сортировка выбором (включает в себя сортировки выбором, пирамидальную, плавную) – сортировка, основанная на обменной сортировке, однако производящая за внутренний цикл только 1 обмен.

Сортировка слиянием – сортировка, разделяющая исходный массив на подмассивы и слияние их обратно в 1 массив в упорядоченном порядке.

Прочие сортировки включают топологические, сортировки сети, битонную и другие.

Основные алгоритмы сортировок. Быстрая сортировка – широко известный алгоритм сортировки, разработанный английским информатиком Чарльзом Хоаром. Один из самых быстрых известных универсальных алгоритмов сортировки массивов. QuickSort является улучшенным вариантом алгоритма сортировки с помощью прямого обмена, известного, в том числе, своей низкой эффективностью. Принципиальное отличие состоит в том, что в первую очередь производятся перестановки на наибольшем возможном расстоянии и после каждого прохода элементы делятся на две независимые группы.

Алгоритм состоит из трёх шагов:

- 1) Выбрать элемент из массива. Назовём его опорным.
- 2) Разбиение: перераспределение элементов в массиве таким образом, что элементы меньше опорного помещаются перед ним, а больше или равные после.

- 3) Рекурсивно применить первые два шага к двум подмассивам слева и справа от опорного элемента. Рекурсия не применяется к массиву, в котором только один элемент или отсутствуют элементы.

Сортировка вставками – алгоритм сортировки, в котором элементы входной последовательности просматриваются по одному, и

каждый новый поступивший элемент размещается в подходящее место среди ранее упорядоченных элементов

На вход алгоритма подаётся последовательность $\{a_1, a_2, \dots, a_n\}$ чисел. В начальный момент отсортированная последовательность пуста. На каждом шаге алгоритма выбирается один из элементов входных данных и помещается на нужную позицию в уже отсортированной последовательности до тех пор, пока набор входных данных не будет исчерпан.

Данный алгоритм можно ускорить при помощи использования бинарного поиска для нахождения места текущему элементу в отсортированной части.

Пирамидальная сортировка – может рассматриваться как усовершенствованная сортировка пузырьком, в которой элемент всплывает (min-heap) / тонет (max-heap) по многим путям.

Сортировка пирамидой использует бинарное дерево. Бинарное дерево – иерархическая структура данных, в которой каждый узел имеет не более двух потомков.

Удобная структура данных для бинарного дерева — это такой массив, что первый его элемент — элемент в корне, а потомки-элементы с индексами $2i+1$ и $2i+2$.

Алгоритм сортировки будет состоять из нескольких основных шагов:

Выстраиваем дерево так чтобы:

- 1) Обмениваем предков с потомками бинарного дерева не нарушая структуру до тех пор, пока все предки не станут меньше потомков.
- 2) Игнорируем один элемент слева из кучи и перестраиваем дерево.
- 3) И повторяем алгоритм

Сортировка слиянием – алгоритм сортировки, который упорядочивает списки в определённом порядке. Эта сортировка — хороший пример использования принципа «разделяй и властвуй». Суть сортировки заключается в разбиении исходного массива на меньшие до массива с 1 элементом так как такой массив является уже упорядоченным.

Алгоритм сортировки:

- 1) Разбиваем исходный массив на меньшие рекурсивно до тех пор пока длина всех массивов не будет равна 1 (массив из 1 элемента)
- 2) 2 упорядоченных массива соединяются таким образом чтобы получить 1 упорядоченный массив
- 3) Передаем упорядоченный массив в предыдущий вызов функции где снова массив объединяется и так до тех пор пока мы не

получим исходный упорядоченный массив, где: *up* - указатель на массив, который нужно сортировать; *down* - указатель на массив *s*, как минимум, таким же размером как у '*up*', используется как буфер; *left* - левая граница массива, передайте 0, чтобы сортировать массив с начала; *right* - правая граница массива, передайте длину массива - 1, чтобы сортировать массив до последнего элемента.

Топологическая сортировка – один из основных алгоритмов на графах. Задача: указать такой линейный порядок на его вершинах, чтобы любое ребро вело от вершины с меньшим номером к вершине с большим номером. Очевидно, что если в графе есть циклы, то такого порядка не существует. Наиболее простая и быстрая реализация этого алгоритма — с помощью обхода в глубину (DFS). (Алгоритм Тарьяна (1976)) **Цвет**. Во время обхода в глубину используется 3 цвета. Изначально все вершины белые. Когда вершина обнаружена, красим ее в серый цвет. Когда просмотрен список всех смежных с ней вершин, красим ее в черный цвет.

Процедура возвращает true, если граф был топологически отсортирован, иначе возвращается false. **Color** — массив, в котором хранятся цвета вершин (0 — белый, 1 — серый, 2 — черный). *N* — количество вершин. **Edges** — массив списков смежных вершин. **Numbers** — массив, в котором сохраняются новые номера вершин. **Stack** — стек, в котором складываются вершины после их обработки. **Cycle** — принимает значение true, если в графе найден цикл.

Анализ алгоритмов сортировок по памяти и времени

По времени:

Для упорядочения важны худшее, среднее и лучшее поведение алгоритма. Если на вход алгоритму подаётся множество **A**, то обозначим $n = |\mathbf{A}|$. Для типичного алгоритма хорошее поведение — это $O(n \log n)$ и плохое поведение — это $O(n^2)$. Идеальное поведение для упорядочения — $O(n)$. Алгоритмы сортировки, использующие только абстрактную операцию сравнения ключей всегда нуждаются по меньшей мере в сравнениях. Тем не менее, существует алгоритм сортировки Хана (Yijie Han) с вычислительной сложностью $O(n \log \log n \log \log \log n)$, использующий тот факт, что пространство ключей ограничено (он чрезвычайно сложен, а за O -обозначением скрывается весьма большой коэффициент, что делает невозможным его применение в повседневной практике). Также существует понятие сортирующих сетей. Предполагая, что можно одновременно (например, при параллельном вычислении) проводить несколько сравнений, можно отсортировать n чисел за $O(\log^2 n)$ операций. При этом число n должно быть заранее известно.

По памяти:

Ряд алгоритмов требует выделения дополнительной памяти под временное хранение данных. Как правило, эти алгоритмы требуют $O(\log n)$ памяти. При оценке не учитывается место, которое занимает исходный массив и независимые от входной последовательности затраты, например, на хранение кода программы (так как всё это потребляет $O(1)$). Алгоритмы сортировки, не потребляющие дополнительной памяти, относят к сортировкам на месте.

ЛИТЕРАТУРА

1. Алгоритм сортировки [Электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/Алгоритм_сортировки – Дата доступа 05.04.2018

УДК 004.42

Студ. А.Д. Самаль
Науч. рук. доц. Н.В. Пацей
(кафедра программной инженерии, БГТУ)

СИСТЕМА БРОНИРОВАНИЯ БИЛЕТОВ МАРШРУТНОГО ТАКСИ НА ОСНОВЕ МОДЕЛИ ОБСЛУЖИВАНИЯ SAAS

SaaS расшифровывается как software as a service — программное обеспечение как услуга. SaaS — это модель предоставления лицензии на программное обеспечение по подписке. Чаще всего такое ПО — это облачное решение, т. е. находящееся на серверах в интернете.

Пользователь получает доступ к сервису через браузер или по API. При этом его поддержкой целиком занимается поставщик услуги. Упрощенно говоря, модель SaaS это когда клиент работает с готовым решением онлайн. Оплачивает доступ и максимально быстро получает на руки готовый инструмент.

Самый простой пример SaaS — это Google Docs, бесплатный сервис для работы с документами. Никаких носителей, драйверов и установок. Заводите аккаунт Google, переходите по ссылке и работаете с текстами, таблицами и презентациями прямо в браузере. Причем в документах одновременно с вами могут работать и другие коллеги. Уехав в командировку, можно зайти в свой аккаунт с любого устройства и продолжить печатать нужный документ. При этом сохранять нужно только изменения настроек, остальные данные сохраняются автоматически [1].