

[Скачать](#)

Операции: * Вставка * Пересечение (пересечение нескольких наборов) * Разница (разница в нескольких наборах) * Поиск по диапазонам * Получение всех интервалов; Начальный интервал + интервал обновления * Удалить интервалы по диапазону * Пользовательские расчеты границ Реализация: * 3-узловое дерево AVL для хранения интервалов * Сохраняет интервалы с типом члена как «тип интервала», например. пользовательский тип интервала будет определен в interval.h * * Запускает метод makebounds в интервальном типе для генерации range() * Запрос интервала вызовет метод границ интервала, который будет рассматривать все интервалы как одноуровневые, а затем использовать двоичный поиск для поиска наименьшего интервала, соответствующего запросу. Интерфейс: * станд::набор * станд::мультимножество * станд::карта * станд::мультикарта * std::unordered_set * std::unordered_multiset * std::unordered_map * std::unordered_multimap * станд::список * станд::дек * std::forward_list * станд::очередь * станд::стек * std::priority_queue * std::set_difference * std::set_intersection * станд::сет_симметричная_разница * std::multiset_difference * std::multiset_intersection * std::multiset_symmetric_difference * std::multimap_difference * std::multimap_intersection * std::multimap_symmetric_difference * std::unordered_set_difference * std::unordered_set_intersection * std::unordered_set_symmetric_difference * std::unordered_multiset_difference * std::unordered_multiset_intersection * std::unordered_multiset_symmetric_difference * std::set_union * std::set_difference_update * std::set_intersection_update * std::set_symmetric_difference_update * std::unordered_set_union * std::unordered_set_difference_update * std::unordered_set_intersection_update

Простой в использовании класс, реализующий класс `\ref IntervalDictionary`. `IntervalDictionary` – это специализированная коллекция для быстрого и эффективного извлечения значений, связанных с интервалами, а не ключей. Он прост в использовании и имеет стандартный интерфейс сбора, и по-прежнему имеют эту возможность интервального поиска. `IntervalDictionary` основан на алгоритме AVL-дерева, который представляет собой сбалансированное двоичное дерево поиска; в основном это решает проблему эффективного поиска значения в отсортированном массиве. В общем случае бинарные деревья поиска могут выполнять $O(\log n)$ операций

поиска. Для плотных структур обход дерева составляет $O(n)$, а сложность большого O — $O(n \log n)$. Когда порядок ввода неизвестен и может быть не полностью отсортирован, внутренняя структура дерева более сложная. Чтобы сохранить постоянную сложность большого O общего бинарного дерева поиска, сбалансированное AVL-дерево приближается к оптимальному поведению $O(\log n)$. Построение AVL-дерева сохраняет свойство, в котором: $[n^{i-1} \leq 2^i + n^{i-1}] [1 \leq i \leq \log_2 n]$ AVL-дерево является оптимальным с точки зрения сложности в наихудшем случае при сохранении операций поиска $O(\log n)$. Структура дерева AVL обеспечивает следующие преимущества: - однократный обход всего дерева; - сложность в наихудшем случае равна $O(\log n)$; - он обеспечивает отношения упорядочения вставки, удаления и включения. Концепция сбалансированного дерева AVL: Любое бинарное дерево поиска является сбалансированным деревом. Коэффициент баланса определяется как `BalanceRatio` и может быть рассчитан как: $[r = \frac{\max(n_r, n_l)}{\min(n_r, n_l)} = \frac{\max(n, n_l)}{\min(n, n_l)}]$ где n — количество значений в дереве и $\{n_r\}$ и $\{n_l\}$ — количество значений справа и слева от корня соответственно. Мы определяем коэффициент баланса с помощью `\href{'s_algorithm}{алгоритма Флойда}`, который представляет собой более простой подход с целым числом шагов, 1709e42c4c

=====
 Интервальный словарь предоставляет специализированную структуру данных на основе интервалов для быстрого и эффективного поиска и поиска. Ключ представляет собой интервал последовательности значений. Эта коллекция подходит для хранения данных, которые включают в себя своего рода интервалы: например, возраст человека, вес яблока или продолжительность проекта. Interval Dictionary поддерживает следующие функции: - Коллекции нескольких типов интервалов - Вставка, удаление и обновление - Запрос относительно типа интервала и границ - Сравнения и равенство - Преобразование между интервалами - Неудовлетворенность - Итерация и поиск
 Интервальный словарь имеет следующие преимущества: - Данные хранятся в виде двусвязного списка, поэтому размер может быть уменьшен до $O(1)$ для вставки и запроса. - Сложность пространства для словаря также составляет $O(1)$ для вставки и запроса. - Поиск быстрее, чем словарь в отношении проблемы поиска. - Реализация этой идеи была оптимизирована с упором на эффективность. - Реализация этой идеи была оптимизирована по производительности. Если вы хотите сравнить с другими структурами интервальных данных, такими как `tr-interval-dictionary`, это должно быть более полезным: - Только сопоставимые типы поддерживаются структурами данных на основе типов (включая поддержку операций порядка) - Количество интервалов ограничено фиксированной верхней границей - Тип ключа - интервалы или интервалы с известной верхней границей `tr-интервал-словарь`: [зависимости]: [документы-url]: [АВТОР]: Эрик Хазелет [API]: [ЛИЦЕНЗИЯ]: [ССЫЛКИ]: [РУКОВОДСТВО]: [ТУТ

What's New In Interval Dictionary?

Получить элементы в коллекции интервалов, которые удовлетворяют свойству. `` `окамл пусть `interval_subset` = "интервал_подмножество" ;; пусть `interval_diff` = "интервал_дифф" ;; пусть `interval_union` = "interval_union" ;; пусть `interval_intersect` = "интервал_пересечение" ;; пусть `interval_domain` = "интервал_домен" ;; пусть `interval_invert`

ДиректХ: 11 ЦП: 4-ядерный ЦП или лучше Оперативная память: 8 ГБ
Хранилище: 30 ГБ свободного места (из них 7 ГБ должно быть доступно при
установке игры) Дополнительные примечания: Вам также потребуется
подключение к Интернету. В игре нет оффлайн игры. Графические карты,
поддерживающие тесселяцию, рекомендуются, но не являются
обязательными. Все остальные функции и технические особенности игры
являются необязательными и могут быть настроены по вашему желанию.
Gears of War 4 будет использовать DirectX 11 для