Дерево – структура данных, представляющая собой совокупность элементов и отношений, образующих иерархическую структуру этих элементов.

Каждый элемент – вершина (узел)

Вершины соединены ветвями

Начальный узел дерева – корень – нулевой уровень

Лист – вершина, в которую входит одна дуга, но ни одна не выходит

Свойства:

Существует узел, в которой не входит ни одной дуги, – корень

В каждую вершину, кроме корня, входит одна дуга

Классификация:

По максимальному количеству потомков у одного узла:

Бинарные – 2 потомка

Тернарные – 3 потомка

М-арные – М потомков

По структуре:

Симметричные

Сбалансированные

Несбалансированные

Полные

Вырожденные

По характеру данных:

С упорядоченными данными

С неупорядоченными данными

Виды:

Бинарные деревья поиска

Строгие \ нестрогие

Полные \ неполные

АВЛ-деревья

Красно-черные деревья

Сильноветвящиеся деревья (В-деревья)

Деревья выражений (дерево синтаксического разбора)

Деревья отрезков

Основные операции с бинарными деревьями:

Создание

Печать

Обход

Вставка элемента

Удаление элемента

Проверка пустоты

Удаление дерева

Обходы деревьев:

Прямой (в глубину) (узел – лево – право)

Симметричный (лево – узел – право)

Обратный (лево – право – узел)

Двоичное дерево поиска упорядочено, если для любой его вершины х справедливы такие свойства:

Все элементы в левом поддереве меньше элемента, хранимого в х

Все элементы в правом поддереве больше элемента, хранимого в х

Все элементы дерева различны

Удаление элемента

Лист – просто удалить и обнулить указатель

Вершина имеет одну ветвь – поставить на его место потомка

Вершина имеет две ветви – поставить самое левое из правого поддерева или самое правое из левого

23.03.2022

AVL-деревья

Сбалансированными деревьями поиска называются деревья, в которых высота левого и правого поддеревьев любого узла отличается не более, чем на n единиц

AVL-деревья, красно-черные деревья, B-деревья

AVL-деревья – n = 1

В случае, если при вставке или удалении узла нарушается сбалансированность дерева, выполняется его балансировка

В AVL-дереве коэффициент сбалансированности любого узла (высота дерева) может принимать значение -1, 0 или 1

Высота узла – длина наибольшего пути от него до листа

Высота листа = 0

Высота пустого дерева = -1

Balance(t) = Height(Left) – Height(Right)

После добавления нового узла необходимо обновить коэффициенты сбалансированности родительских узлов

Если в родительском узле коэффициент сбалансированности стал равным 2 или -2, необходимо выполнить балансировку с помощью поворотов

Одиночный правый поворот

Одиночный левый поворот

Двойной лево-правый поворот

Выполняется после добавления элемента в правое поддерево левого дочернего узла дерева

Двойной право-левый поворот

Красно-черные деревья относятся к сбалансированным бинарным деревьям поиска

Каждый узел хранит дополнительное поле color, обозначающее цвет: красный или черный, и для которых выполнены приведенные ниже свойства

Будем считать, что если left или right равны null, то это «указатели» на фиктивные листья. Таким образом, все узлы – внутренние (нелистовые)

Свойства:

1. Каждый узел либо красный, либо черный
2. Каждый лист (фиктивный) – черный
3. Если узел красный, то оба его сына – черные
4. Все пути, идущие от корня к любому фиктивному листу, содержат одинаковое количество черных узлов (глубина по черным узлам)
5. Корень – черный

Грубая оценка правил 3 и 4 показывает, что длины двух высот соседних поддеревьев отличаются не более, чем в 2 раза

Каждый новый узел изначально считается красным. Если это нарушает одно из правил, обычно 3 и 4, то производится балансировка

Черной высотой узла называется количество черных узлов на пути от этого узла к узлу, у которого оба сына – фиктивные листья

Черная высота дерева – черная высота его корня

Вставка узла:

Сначала узел добавляется в дерево с помощью стандартного алгоритма вставки узла в двоичное дерево поиска

Вновь добавленный узел красится в красный цвет

Если это первый узел в дереве, то он становится корнем и перекрашивается в черный цвет

Далее производится проверка, не нарушились ли свойства КЧ-дерева

Если добавленный узел не первый, то он красный, поэтому свойство 4 об одинаковом количестве черных узлов на любом пути от корня к листу, не нарушается

Если родитель нового узла черный, то свойство 3 о том, что если узел красный, то оба его сына черные, также не нарушается

Но если родитель нового узла красный, то это свойство будет нарушено – возникает так называемое красно-красное нарушение

Тогда потребуется перекраска и, возможно, перестройка дерева

Балансировка при добавлении

Случай 1 – отец и дядя – красные

Достаточно выполнить перекраску расположенных выше узлов

Корень перекрашивается в черный цвет

Случай 2 – отец красный, дядя черный

При этом цепочка узлов образует прямую линию

Потребуется одинарный поворот деда относительно отца и перекраска

Случай 3 – отец красный, дядя черный

При этом цепочка узлов образует угол

Потребуется двойной поворот из раздела АВЛ

Случаи 2 и 3 являются терминальными, а рекурсивно продолжиться наверх может только процедура в случае 1, а она требует поворотов

27.04.2022

Разреженные матрицы

Матрица называется разреженной, если в ней «много» нулевых элементов

Критерием разреженности матрицы считается число ее ненулевых элементов

Чтобы матрица порядка n была разреженной, число ее ненулевых элементов должно выражаться как n1+g, где g < 1

Диагональная схема хранения ленточных матриц – хранят симметричные диагонали

Профильная схема хранения симметричных матриц – массив преддиагональных чисел и массив индексов диагональных элементов

Схема Кнута – хранение элементов и индексов, номеров следующих элементов, точек входа

Кольцевая КРМ-схема – схема Кнута без хранения индексов

Разреженный строчный формат