Древовидные структуры данных: B-деревья, R-деревья, Интервальные деревья. Представление графов.

С.Ю. Сартасов Д. В. Луцив

Кафедра системого программирования СПбГУ

CS103

Содержание

B-дерево. Термины и определения

Придумали Р. Байер и Э. Мак-Крейт, 1972 (М. Кауфман – не опубликовал)
B-дерево порядка $2t$ (по БМК) или $2t+1$ (по Кнуту) удовлетворяет условиям:

- В листьях дерева хранится $[t..2t]$ числа.
- В узлах дерева хранятся кроме этого ссылки на $[t+1..2t+1]$ поддерева.

Замечания

Расхождения в аксиоматике:

- Р. Байер и Э. Мак-Крейт в исходной статье листьями называли нижний уровень вершин с ключами, соответственно с другой аксиоматикой. Пустые листья предложил Д. Э. Кнут.
- Некоторые определения оперируют количеством ключей, а не детей.

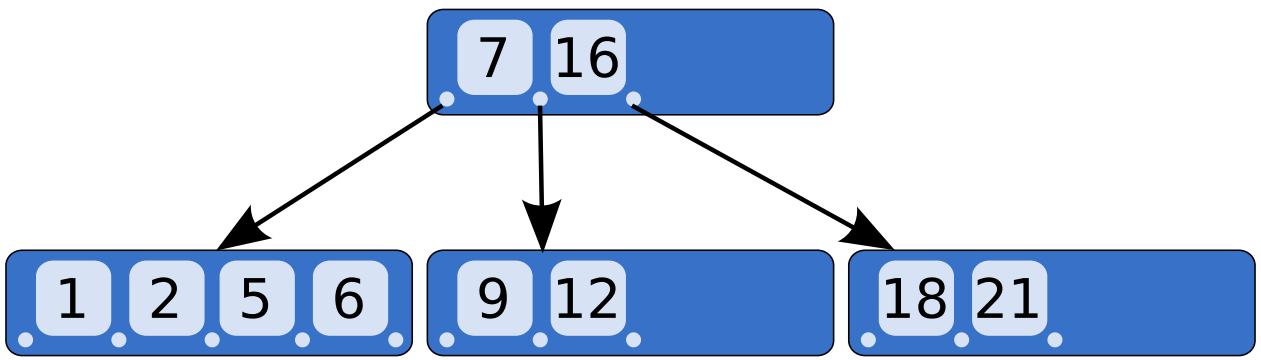


Рис.: Порядок 2 (БМК 1972) или 5 (К 1998) ▸ Википедия

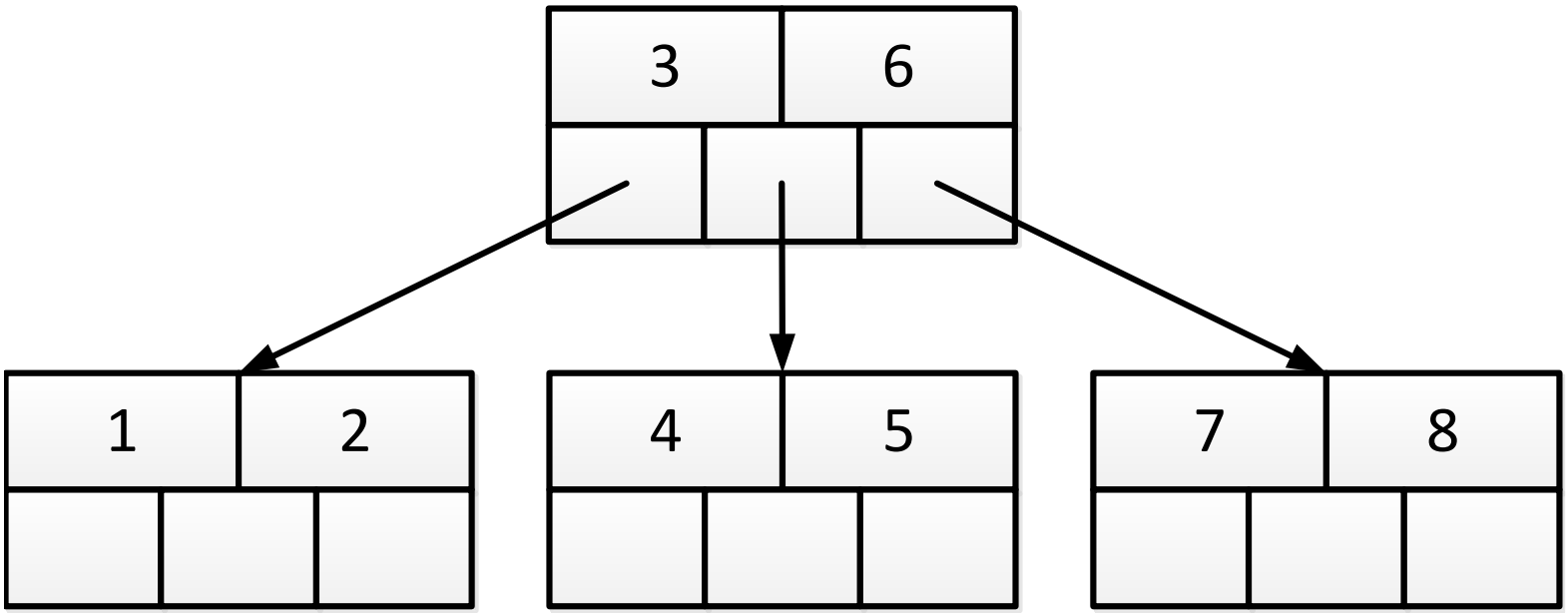
Свойства B-дерева:

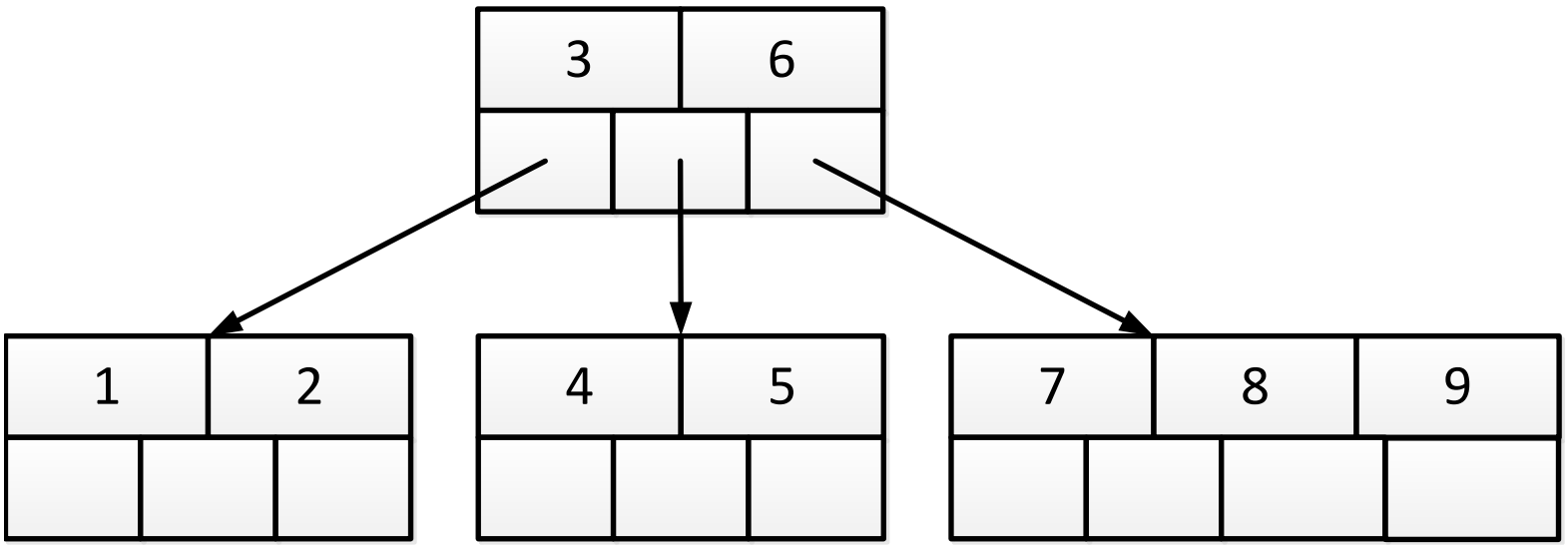
- Сильно ветвистое (t велико)
- Идеально сбалансировано по высоте
- Внутренние узлы заполнены минимум наполовину

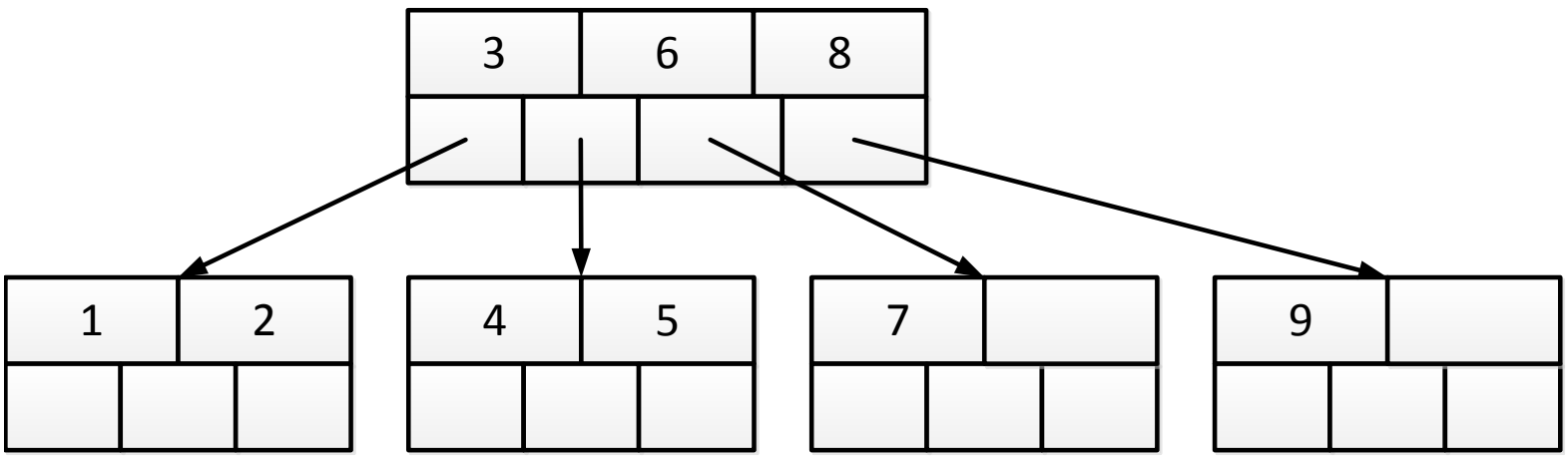
Используется при организации индексов. Нижние уровни часто выносятся во внешнюю память.

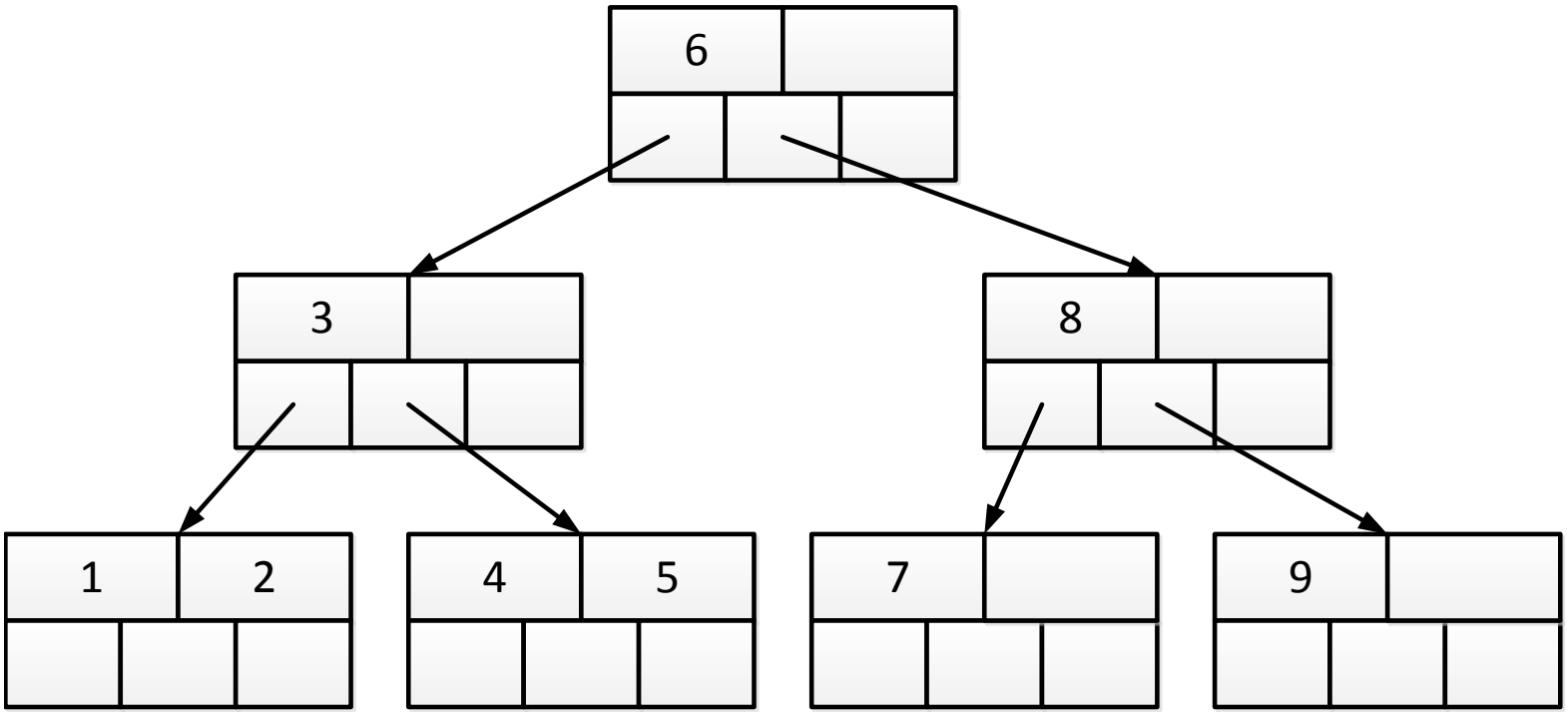
Алгоритм вставки

- Добавить узел в соответствующее место в соответствующем листе.
- В случае переполнения - разделить узел на два, число посередине вставить в родительский узел. Повторять до корня.


3
6
1
2
4
5
7
8


3
6
1
2
4
5
7
8
9


3
6
8
1
2
4
5
7
9


6
3
8
1
2
4
5
7
9

Оценки сложности:

- Верхняя оценка высоты - $h \leq \lfloor \log_t((N+1)/2)+1 \rfloor$
- Время поиска $O(\log_2 t \log_t N)$
- Время вставки $O(\log_2 t \log_t N)$

Удаление

Пример с переливанием

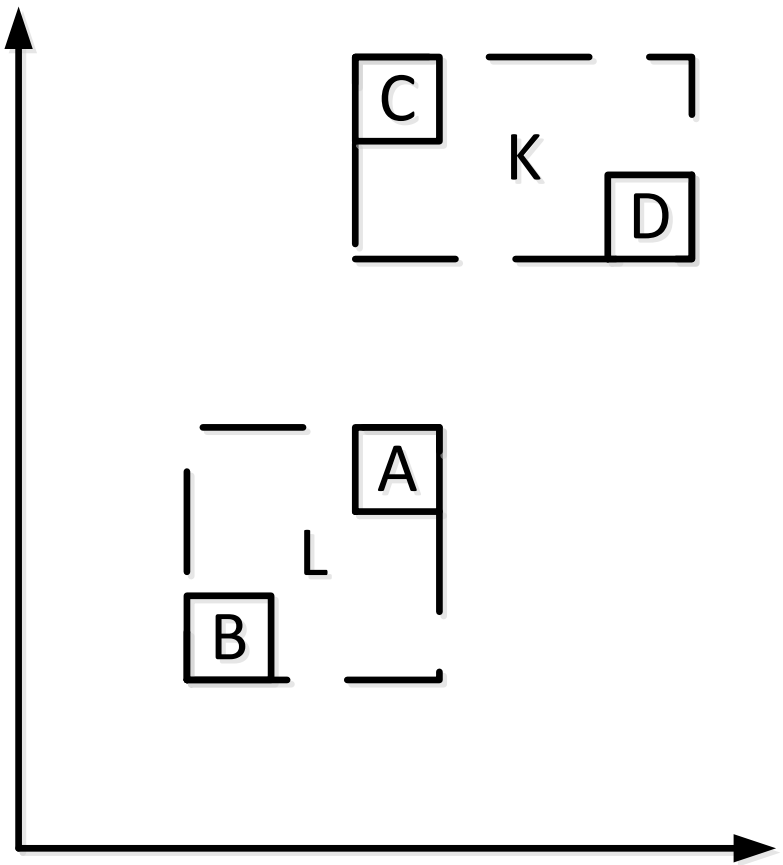
R-дерево. Термины и определения

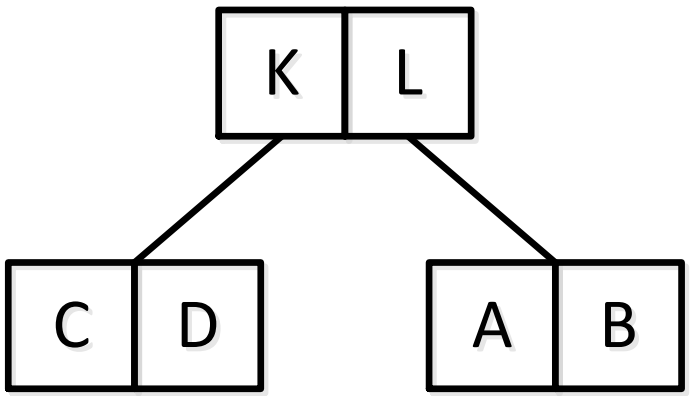
R-дерево — обобщение B-дерева на случай многомерных объектов. Для двумерного случая — фигуры на плоскости, отрезки, точки.
Максимальное количество элементов в узле - $M \geq 2$
Минимальное количество элементов в узле - $m \leq M/2$

Организация:

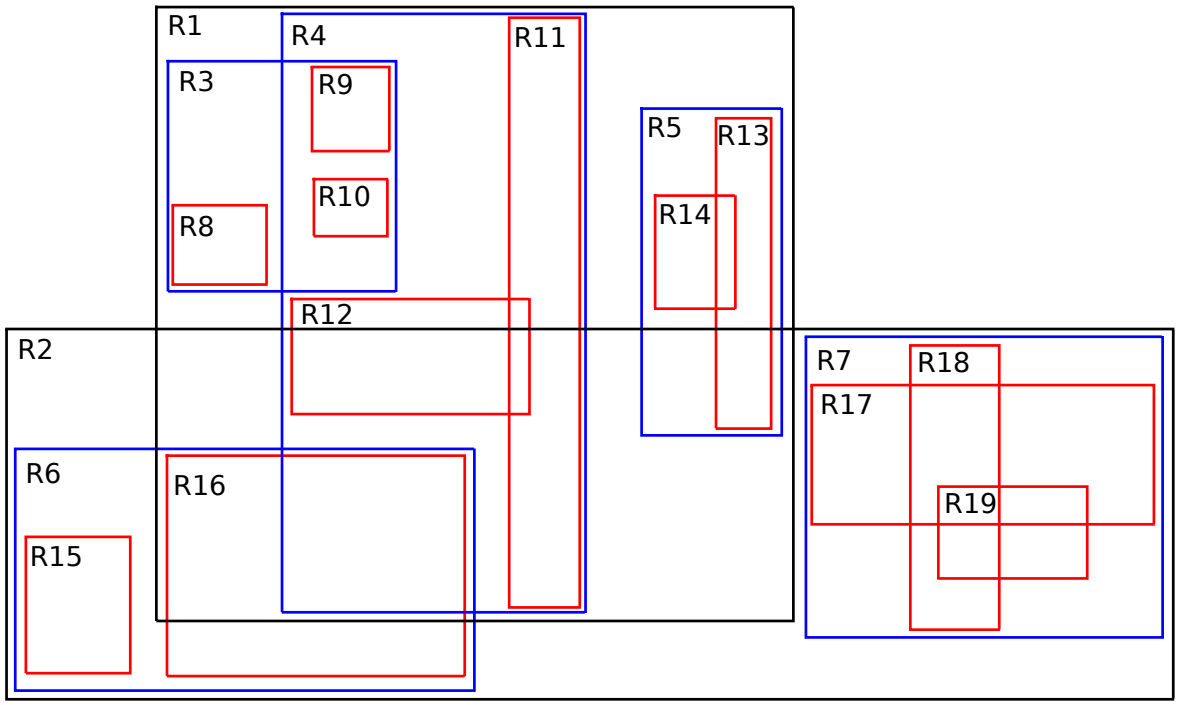
- Запись в узле дерева — минимальный прямоугольник со сторонами, параллельными осям координат, в который можно вписать хранимый(е) ниже объект(ы).
- В корне — минимум 2 объекта (если он не является листом, тогда, возможно, меньше)
- В листе — от m до M записей (если он не является корнем)
- В промежуточном узле — от m до M записей

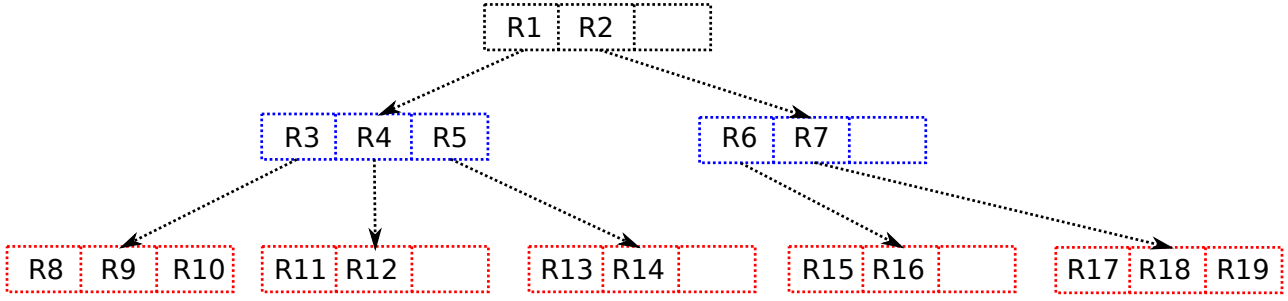
В идеале





В жизни


R1
R4
R11
R3
R9
R5
R13
R10
R14
R8
R12
R2
R7
R18
R17
R6
R16
R19
R15


R1
R2
R3
R4
R5
R6
R7
R8
R9
R10
R11
R12
R13
R14
R15
R16
R17
R18
R19

Свойства:

- Сильно ветвистое
- Сбалансированное по высоте
- Вхождение в прямоугольник не гарантирует положительного результата поиска
- Возможное пересечение прямоугольников внутри одного элемента — неоднозначность поиска

Поиск объектов, входящих в данную область. Очень упрощённо — Определение объекта экранной формы, на который щёлкнули мышкой

Алгоритм вставки

Похож на B-дерево.
Упрощённый вариант разделения.

- Взять наибольшее измерение параллелотопа.
- Найти «центр масс» по этому измерению.
- Разделить по центру масс.

Описание и построение интервального дерева

Похоже на R-дерево

1. Дано множество интервалов
2. Нахдится «центр масс» множества
3. Интервалы строго слева от центра идут в левое поддерево
4. Интервалы строго справа — в правое
5. Список оставшихся (включающих центр) остаётся в узле
6. Левое и правое поддеревья рекурсивно строятся для соответствующих подмножеств, пока они не пусты

▸ Дерево интервалов

Описание и построение интервального дерева

Похоже на R-дерево

1. Дано множество интервалов
2. Нахдится «центр масс» множества
3. Интервалы строго слева от центра идут в левое поддерево
4. Интервалы строго справа — в правое
5. Список оставшихся (включающих центр) остаётся в узле
6. Левое и правое поддеревья рекурсивно строятся для соответствующих подмножеств, пока они не пусты

▸ Дерево интервалов

Поиск в интервальном дереве

При поиске точки, она ищется в вершине и в одном из поддеревьев.
При поиске интервала ищется, с какими интервалами он пересекаются, рекурсивный спуск с учётом его границ.

Описание

- Каждый узел хранит один символ и ссылки на трёх детей — младшего, старшего и среднего:
  - у младшего ребёнка символ строго перед символом вершины;
  - у старшего — строго после;
  - у среднего — продолжение строки.
- Каждый узел хранит либо признак конца строки, либо ссылку, возможно пустую, на значение, поставленное в соответствие ключу.

▸ Троичное дерево поиска

Поиск и вставка

Ищем строку:

- Если символ в вершине совпал с очередным символом строки, продолжаем в среднем поддереве со следущего символа строки
- Если меньше или больше, то по строке не двигаемся, а идём в поддеревья

Вставляем строку:

- Пока у вершин есть дети, ищем очередные символы строки
- Когда дошли до листа, начинаем наращивать средними детьми

Поиск и вставка

Ищем строку:

- Если символ в вершине совпал с очередным символом строки, продолжаем в среднем поддереве со следущего символа строки
- Если меньше или больше, то по строке не двигаемся, а идём в поддеревья

Вставляем строку:

- Пока у вершин есть дети, ищем очередные символы строки
- Когда дошли до листа, начинаем наращивать средними детьми

Префиксное дерево

Для хранения строк и связанных с ними значений.
В каждой вершине — тем или иным способом реализованный словарь, позволяющий по следующему символу строки получить поддерево для продолжения поиска.

▸ Префиксное дерево

Представление графов

Зависит от вида графа

- Наивное
- Двумерный массив
- Одномерный массив

Упражнение

Найти эффективный способ хранить треугольный двумерный массив, как одномерный

Представление графов

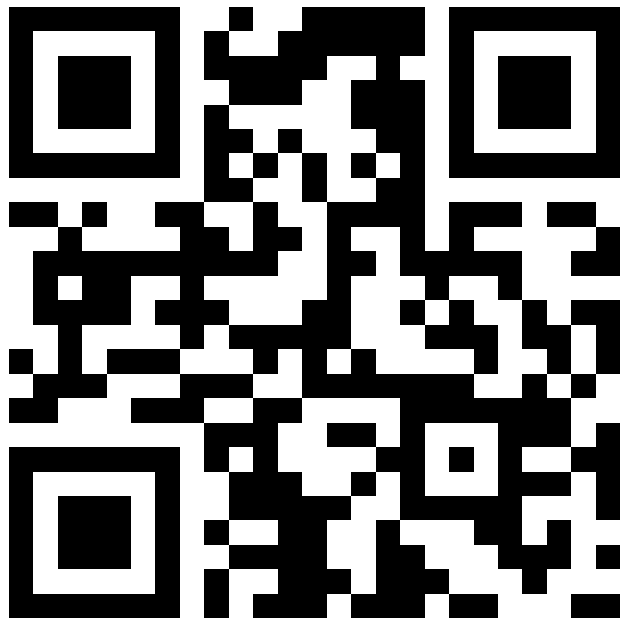
Зависит от вида графа

- Наивное
- Двумерный массив
- Одномерный массив

Упражнение

Найти эффективный способ хранить треугольный двумерный массив, как одномерный

Вопросы



EDU.DLUCIV.NAME