Тема: Двоичное Б-дерево поиска (ДБД)

Цель работы: Изучение процесса программного построения ДБД.

1. Постановка задачи

1. Разработать подпрограмму построения ДБ-дерева для массива целых чисел

2. Построить ДБ-дерево из 100, 200,…, 500 вершин (данные в вершинах произвольные, но все различные). Распечатать обход дерева слева направо.

3. Для построенного ДБ-дерева вычислить размер, контрольную сумму, высоту и среднюю высоту (как для двоичного дерева) и высоту ДБ-дерева как количество уровней, сравнить их с аналогичными характеристиками АВЛ-дерева. ДБ-дерево необходимо строить для той же последовательности данных, что и АВЛ-дерево. Заполнить таблицу и проанализировать полученные результаты.

2. Описание используемых алгоритмов

Деревья, имеющие вершины со многими потомками, будем называть сильноветвящимися.

Б – дерево порядка m – это дерево со следующими свойствами:

1. В каждой странице хранится k элементов данных d1 < d2 < ... < dk и k+1 указатель p0, p1, ...pk. Каждый указатель pi либо равен NIL, либо указывает на вершину, все элементы которой больше di, но меньше di+1.

Построение Б-дерева или включение нового элемента данных D в Б-дерево происходит следующим образом:

• Выполним поиск элемента D в дереве.

• Если элемента D нет в дереве, то мы имеем страницу a и позицию R, в которой ожидали найти элемент D.

• Вставим элемент в позицию R+1, при этом количество элементов на странице k увеличилось на 1.

• Если k < = 2m, то процесс включения закончен.

• Если k > 2m (переполнение страницы), то создаём новую страницу b, переносим в неё m правых элементов из страницы a, а средний элемент переносим на один уровень вверх на родительскую страницу.

• Включение элемента в родительскую страницу производится по такому же алгоритму.

• Если родительской страницы нет, то она создаётся и в неё включается один элемент.

Эта схема сохраняет все характерные свойства Б-дерева. Получившиеся две новые страницы содержат ровно по m элементов. Включение элемента в родительскую страницу может опять перевести к переполнению, то есть разделение страниц может распространиться до самого корня. В этом случае может увеличиться высота дерева. Таким образом, Б-деревья растут обратно: от листа к корню.

Двоичное Б-дерево состоит из вершин (страниц) с одним или двумя элементами. Следовательно, каждая страница содержит две или три ссылки на поддеревья.

Двоичные Б-деревья представляют собой альтернативу АВЛ-деревьям. При этом поиск в двоичном Б-дереве происходит как в обычном двоичном дереве.

Высота двоичного Б-дерева . Если рассматривать двоичное Б-дерево как обычное двоичное дерево, то его высота может увеличиться вдвое, т.е.  .

Построение двоичного Б-дерева происходит путем добавления новой вершины в уже существующее дерево. Введем логическую переменную VR, показывающую вертикальный рост дерева (в случае, если страница переполнилась и средний элемент передается на вышележащий уровень) и логическую переменную HR, определяющую горизонтальный рост дерева (если новый элемент размещается на этой же условной странице). Также определим показатель баланса BAL для каждой вершины, который принимает значение 0, если у данной вершины есть только вертикальные ссылки (вершина одна на странице), и значение 1, если у данной вершины есть правая горизонтальная ссылка.

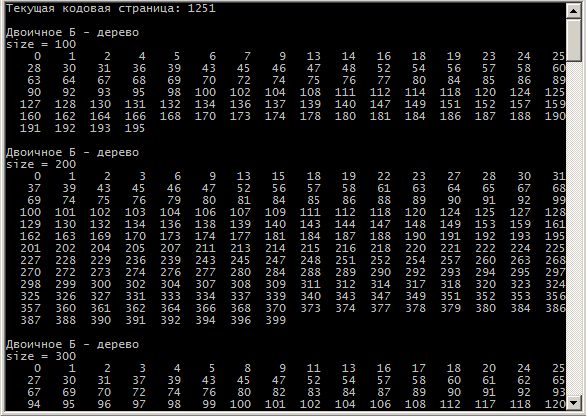
При добавлении элементов в двоичное Б-дерево различают 4 ситуации, возникающих при росте левых или правых поддеревьев. Самый простой случай) — рост правого поддерева вершины А, когда А — единственный элемент на странице. Тогда вертикальная ссылка просто превращается в горизонтальную (HR=1, баланс вершины А равен 1). Если на странице уже два элемента (2), то при добавлении новой вершины С средняя вершина В передается на вышестоящий уровень (VR=1, баланс вершины В равен 0).

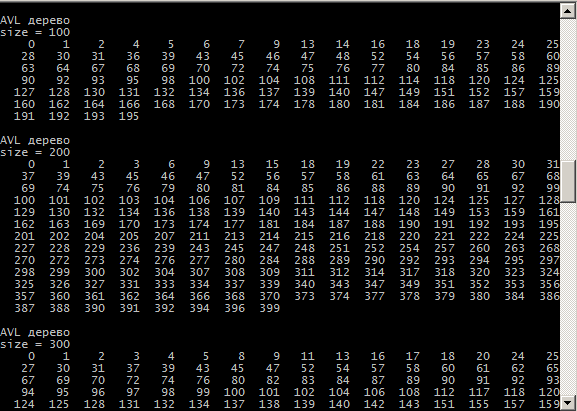
В случае роста левого поддерева, если вершина В одна на странице, то вершина А добавляется на эту страницу. Однако левая ссылка не может быть горизонтальной, поэтому требуется переопределение ссылок (HR=1, баланс вершины А равен 1). Если на странице два элемента, то, как и в случае 2, средняя вершина В поднимается на вышестоящий уровень (VR=1, баланс вершины В равен 0).

При построении двоичного Б-дерева реже приходится переставлять вершины, поэтому АВЛ-деревья предпочтительней в тех случаях, когда поиск ключей происходит значительно чаще, чем добавление новых элементов. Кроме того, существует зависимость от особенностей реализации, поэтому вопрос о применение того или иного тапа деревьев следует решать индивидуально для каждого конкретного случая.

3. Результат работы программы

Фрагменты работы программы показаны на рисунках.





Результаты замеров для ДБ-дерева записываются в файл DBD.txt.

Результаты замеров для АВЛ-дерева записываются в файл AVL.txt.

4. Анализ и сравнение полученных результатов с теоретическими оценками

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Размер дерева | АВЛ-дерево | | | ДБД | | | |
| Контр.  сумма | Высота фактическая | Теор. оценки для сред. высоты | Контр.  сумма | Кол-во уровней | Теор. оценки для высоты ДБД | Теор. оценки для сред. высоты двоичного дерева |
| 100 | 9652 | 8 | 5.04 | 9652 | 10 | 10 | 5.56 |
| 200 | 40815 | 11 | 6.29 | 40815 | 12 | 12 | 6.51 |
| 300 | 88492 | 11 | 6.63 | 88492 | 14 | 13 | 7.69 |
| 400 | 163252 | 11 | 7.03 | 163252 | 15 | 13 | 8.50 |
| 500 | 246355 | 12 | 7.35 | 246355 | 17 | 14 | 9.19 |

Как видно из результатов замеров высота ДБД несколько больше, чем высота АВЛ-дерево для одних и тех же данных.