

Лабораторная работа №3

Бинарные деревья

Задание

1. Ознакомиться с постановкой задачи
2. Написать программу для ее реализации
3. Разработать тесты
4. Исходные данные поместить в файл input.dat
5. Результаты вывести на экран. Исходные данные и результаты вывести также в выходной файл output.dat
6. Предусмотреть вывод исходного и выходного деревьев в табличной форме или в графике на экран. Вывод в табличной форме означает вывод в прямом порядке, т.е. Корень, Левое поддерево, Правое поддерево. Вывести деревья в выходной файл.
7. Оформить отчет

Варианты заданий

1. Проверить, содержится ли второе дерево как поддерево в первом дереве, и если содержится, определить уровень, на котором находится в первом дереве узел, содержащий ключ, равный ключу корня второго дерева.
2. На основании исходных данных сформировать бинарное дерево. Определить максимальное число потомков одного узла в исходном дереве на двух ближайших уровнях(дети и внуки). Указать, на каком уровне находится узел, число потомков которого на этих уровнях максимально
3. Удалить вершину с заданным значением ключа из бинарного дерева поиска. Предусмотреть вывод дерева сразу после его формирования и после удаления элемента.
4. Проверить два бинарных дерева на информационную эквивалентность, т.е. убедиться, что все данные первого дерева содержатся во втором и все данные второго дерева содержатся в первом. Вывести информацию о информационной эквивалентности или ее отсутствии.
5. Занести в каждый узел бинарного дерева номер его уровня, т.е. при формировании дерева из данных исходного файла формируются информационные поля узла и дополнительное поле, содержащее номер уровня узла. При выводе дерева предусмотреть печать ключевого поля и номера уровня узла.
7. Восстановить неупорядоченное бинарное дерево по известным порядкам его обходов: прямому и симметричному.
8. Найти высоту сформированного на основании данных, содержащихся во входном файле бинарного дерева.
9. Сформировать два бинарных дерева. Удалить из второго бинарного дерева информацию, содержащуюся в обоих деревьях.
10. Получить копию сформированного бинарного дерева
11. Найти в бинарном дереве узел с заданным значением ключа и удалить его правого потомка.
12. Подсчитать число листьев в бинарном дереве.
13. Добавить во второе бинарное дерево ту информацию из первого, которая во втором дереве ранее не содержалась.
14. . Удалить из сформированного бинарного дерева узлы с информационными полями, принадлежащими [a,b]. Информационные поля – числовые.
15. Обойти бинарное дерево в обратном порядке.
16. Написать программу для получения всех элементов, размещенных на дереве сортировок с признаками вне заданного диапазона.
17. Написать программу для определения среднего числа сравнений для доступа к любому элементу заданного дерева.

18. Получить скобочное представление бинарного дерева, <ключ корня>(< левый потомок>,< правый потомок>). Потомки также должны быть представлены скобочным представлением
19. Сформировано упорядоченное БД. Каждый узел дерева соответствует файлу и содержит имя файла и дату последнего обращения к нему. Написать программу, которая обходит дерево и удаляет все те файлы, обращение к которым происходило до определенной даты. Ключ – имя файла.
20. Написать программу для поиска вершины с ключом, который непосредственно предшествует заданному. Удалить найденный элемент.
21. Реализовать алгоритм получения упорядоченной последовательности элементов, рассортированных на дереве, а затем построения выровненного дерева по этой последовательности.
22. На основании информации, хранящейся в файле, создается дерево. Каждый узел содержит ключевое поле и поле счетчика обращений. Ключ – целое число. Сразу после создания дерева все счетчики нулевые. Частота обращений к каждому узлу измеряется эмпирически. Значение счетчика увеличивается на единицу, если востребована информация из того узла, в котором счетчик находится. Через определенный интервал дерево реорганизуется, при этом в новое дерево узлы добавляются в порядке убывания счетчиков частоты обращений. Выполнить реорганизацию дерева после выполнения $n(n \leq 100)$ операций поиска информации по ключу.
23. Написать программу для поиска элемента с таким ключом, который является минимальным среди ключей элементов дерева, которые больше заданного значения. Удалить найденный узел из дерева.
24. Заданы два дерева. Удалить из второго дерева элементы, ключи которых совпадают с элементами первого и ключевые признаки которых четны.
25. Заданы три дерева. Удалить из второго дерева элементы, ключи которых совпадают с ключами третьего и не совпадают с ключами первого
26. Реализовать удаление всех элементов заданного уровня из дерева.
27. Реализовать удаление элемента с заданным значением ключевого признака если он расположен в дереве на заданном уровне.
28. Найти в дереве узел с заданным значением ключа и удалить те узлы, которые находятся на пути от корня до узла с заданным значением ключа. Корень не удалять.