**Лабораторная работа №3**. **Структуры данных и их применение**.

Решить задачу, в соответствии с вариантом подгруппы. В качестве среды разработки использовать IntelliJ IDEA. В разработке алгоритма использовать строго типизированный объектно-ориентированный язык программирования Java.

1. Организовать хеш-таблицу с открытой адресацией, используя процедуру поиска и вставки по ключу. Для формирования начального хеш-адреса использовать метод деления, а затем при возникновении коллизии процедуру квадратичного исследования.
2. Написать программу, определяющую является ли двоичное дерево деревом поиска.
3. Имеется дерево, корень которого соответствует основателю рода. Сыновья каждой вершины задают сыновей и дочерей соответствующего человека. Указываются имена двух человек (например, А и В) . Сообщить, какая из следующих ситуаций имеет место:  
   1) А предок В;  
   2) В предок А;  
   3) А и В имееют ближайшего общего предка С.
4. При запуске программы в консоли должно открыться меню с выбором "Работа со списками" или "Работа с деревьями". При выборе какого-либо из пунктов должно открыться подменю. Подменю "Работа со списками" должно включать в себя пункты: "Включить новый элемент после i-ого по номеру элемента", "Исключить элемент после i-ого по номеру элемента", "Поменять местами i-ый и i-ый+1 по номерам элементы". Подменю "Работа с деревьями" должно выполнять следующие действия: создание бинарного дерева (дерева поиска) в диалоге с пользователем (способ формирования дерева: с учётом значений ключа), обход бинарного дерева каждым из трёх способов с выдачей на экран содержимого информационных полей, включение элемента в бинарное дерево (согласно алгоритму формирования дерева), удаление заданного узла из дерева (без поддерева), удаление дерева с освобождением памяти, вывод дерева на экран. А так же определить количество листьев на заданном уровне дерева, определить количество узлов (не листьев) бинарного дерева.
5. Составить программу  для построения и обработки дерева общего вида или упорядоченного двоичного дерева, содержащего узлы целого типа. После того как дерево, создано с ним можно выполнять следующие действия:  
   Добавление нового узла (для двоичного дерева положение нового узла определяется в соответствии с требованием сохранения порядка, для дерева общего вида должен задаваться путь до добавляемого узла, добавляемый узел становится младшим сыном).  
   Текстовая визуализация дерева (значение каждого узла выводится на отдельной строке, с отступом пропорциональным глубине узла (шаг отступа равен двум пробелам), в порядке старшинства узлов).  
   Удаление узла (двоичное дерево перестраивается в соответствии с требованием сохранения целостности и порядка; для дерева общего вида удаляется все поддерево, исходящее из удаляемого узла; должно быть предусмотрено корректное освобождение памяти). Определение числа вершин, степень которых совпадает со степенью дерева.
6. Задан граф (орграф) в виде матрицы смежности. Составить программу  
   а) проверки, есть ли в графе петли;  
   б) поиска в графе изолированной вершины (не смежной с дру­гими);  
   в) определения степени графа;  
   г) получения последовательности ребер.
7. Запрограммировать структуры данных и операции над ними для представления графа в виде  
   а) последовательности ребер (дуг),  
   б) векторов смежности,  
   в) матрицы смежности,  
   г) списков смежности,  
   д) матрицы весов,  
   е) сети,  
   ж) матрицы инцидентности,  
   з)списковой структуры.  
   Необходимые детали представления уточнить  самостоятельно.
8. Дан граф (орграф). Составить описание данных для его представления и фрагмент программы  
   а) получения кратчайших путей и расстояний между всеми вер­шинами по алгоритму Флойда и вывода кратчайшего пути от вершины А до вершины В.  
   б) получения матрицы связности по алгоритму Уоршалла и вывода номеров вершин каждой компоненты связности графа.
9. Разработать программу, выполняющую обработку данных на основе рекурсивных алгоритмов.  
   Дано N городов и известны расстояния между ними. Не все города связаны друг с другом дорогой. Найти все возможные маршруты из города А в город В (ни в один город не заходить дважды). Определить самый длинный и самый короткий маршрут.
10. Система двусторонних дорог такова, что для любой пары городов можно указать соединяющий их путь. Найдите такой город, сумма расстояний от которого до остальных городов минимальна.
11. Для заданной системы двусторонних дорог определите, можно ли, закрыв какие-нибудь три дороги, запретить перемещение из города А в город Б.
12. Дана система двусторонних дорог. N-периферией называется множество городов, расстояние от которых до выделенного города (столицы) больше N. Для данного N определите N-периферию.
13. Определите, можно ли в данной системе двусторонних дорог проехать из города А в город B так, чтобы посетить город С и не проезжать ни по какой дороге более одного раза.
14. За проезд по каждой дороге взымается некоторая пошлина. Найдите путь из города А в город B с минимальной величиной S+P, где S - сумма длин дорог пути, а P - сумма пошлин проезжаемых дорог.
15. Найдите медиану графа, то есть такую его вершину, что сумма расстояний от нее до остальных вершин минимальна.