ЛАБОРАТОРНАЯ РАБОТА № 6
ПОИСК КРАТЧАЙШХ ПУТЕМ МЕТОДОМ ФЛОЙДА

Цель работы: Изучить алгоритм поиска кратчайших путей в графе используя алгоритм Флойда.

Теоретические сведения

Двоичное (бинарное) дерево — иерархическая структура данных, в которой каждый узел имеет не более двух потомков.

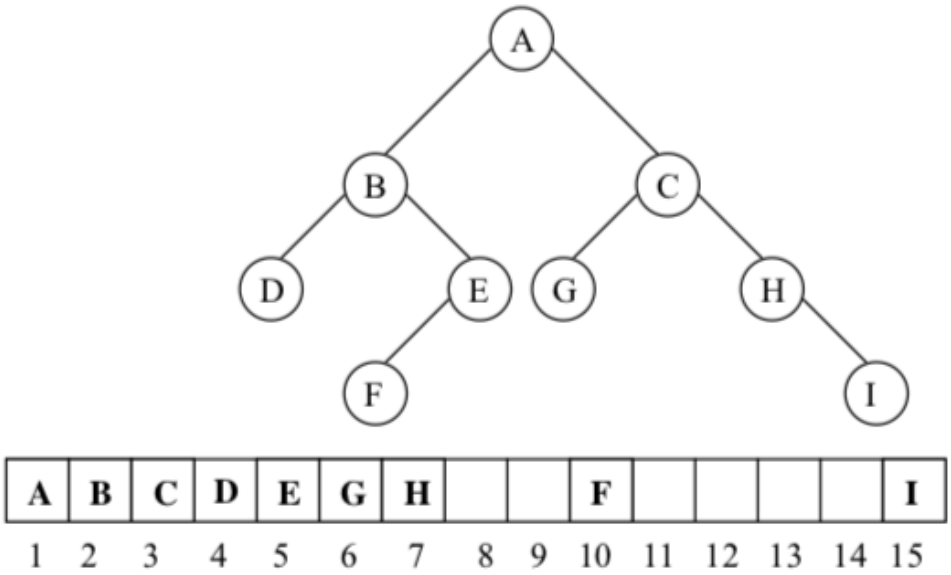
В программировании – структура данных, которая имеет корень и дочерние узлы, без циклических связей.

Если рассмотреть отдельно любой узел с дочерними элементами, то получится тоже дерево. Узел называется внутренним, если имеет хотя бы одно поддерево. Узлы, не имеющие потомков (оба потомка которых равны NULL) называются листьями.

Любой узел не может иметь более двух потомков. Их называют просто — левый и правый потомок, или левое и правое поддерево.

Основное правило его построения – левый потомок меньше текущего узла, а правый потомок больше.

Достоинством представления бинарных деревьев в памяти с последовательной организацией является простота доступа как от предка к потомку, так и от потомка к предку, а недостатком – то, что если дерево не является полным, в массиве появляется большое число пустых элементов. Ограниченный размер массива затрудняет включение в дерево новых узлов, т.к. при этом требуется изменять размер массива. Удаление узла из дерева и соответствующее изменение его структуры также потребует модификации содержимого массива.



Представление бинарного дерева в памяти с последовательной организацией

Пример реализации дерева

Класс TreeNode содержит описание одного узла дерева.

```
public class TreeNode {
  public char info;
  public TreeNode left;
  public TreeNode right;

  public TreeNode () {
    ...
  }
  public TreeNode (char info) {
  ...
}
public TreeNode(char info, TreeNode left, TreeNode
right)
{
  info = info;
  left = left;
  right = right; }
```

```
}
public class BinaryTree {
  public TreeNode root {
  public BinaryTree () {
  root = null;
}
}
```

Дерево задается ссылкой на его корень. Если дерево пусто, ссылка на его корень равна null.

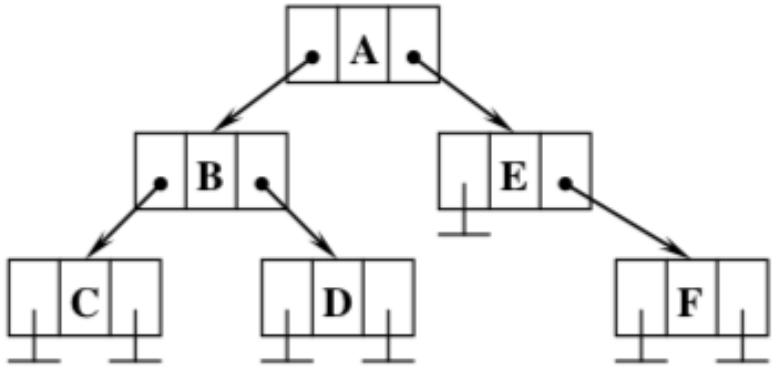


Рисунок – Связное представление бинарного дерева

Наиболее типичная операция, выполняемая над бинарными деревьями, – обход дерева. Обход – это операция, при выполнении которой каждый узел обрабатывается ровно один раз одинаковым образом. Обход дерева заключается в разбиении дерева на корень, левое и правое поддеревья и применении к каждому из поддеревьев соответствующей операции обработки до тех пор, пока в процессе разбиения не будет получено пустое дерево. Полный обход дерева дает линейную расстановку узлов, что облегчает выполнение многих алгоритмов. Существует несколько принципов упорядочения, которые естественно вытекают из структуры дерева. Как и саму древовидную структуру, их удобно представить с помощью рекурсии. Различным принципам упорядочения соответствуют три левосторонних

алгоритма обхода в глубину (и три симметричных правосторонних алгоритма): нисходящий, восходящий, смешанный, а также обход в ширину.

Нисходящий (прямой) обход выполняется согласно алгоритму “корень-левый-правый” (К-Л-П):

1. обработать корневой узел;
2. обойти левое поддерево в нисходящем порядке;
3. обойти правое поддерево в нисходящем порядке.

```
public void KLP(TreeNode root)
{
  if ( root !=null )
  {
    Console.WriteLine(root.info );
    KLP(root.left );
    KLP(root.light )
  }
}
```

Задание на лабораторную работу

1. Составить программу, осуществляющую ввод двоичного дерева с числом уровней не менее 4х.
2. Выполнить обход дерева и вывести на экран его узлы