Отчет 1. Деревья. Терминология деревьев. Классификация деревьев. Представление деревьев.

Терминология деревьев

Дерево – одна из наиболее широко распространенных структур данных в информатике, эмулирующая древовидную структуру в виде набора связанных узлов.

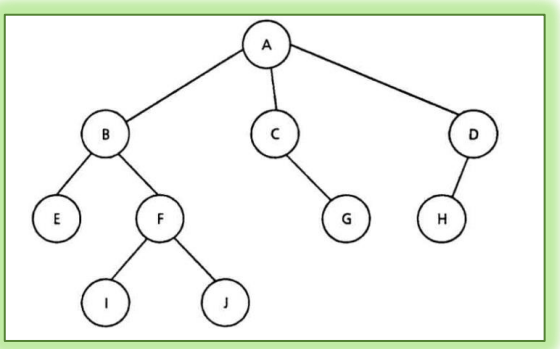
Древовидная структура характеризуется множеством ***узлов***, происходящих от единственного начального узла, называемого корнем. В терминах генеалогического дерева узел можно считать ***родителем***, указывающим на 0, 1 или более узлов, называемых ***сыновьями***. Дерево может представлять несколько поколений семей. Сыновья узла и сыновья их сыновей называются ***потомками***, а родители и прародители – ***предками*** этого узла. Каждый некорневой узел имеет только одного родителя, и каждый родитель имеет 0 или более сыновей. Узел, не имеющий детей, называется ***листом***.

На рисунке:

A – корень

B – родитель сыновей E и F

E, F, I, J – потомки узла B



Каждый узел является корнем ***поддерева***, которое определяется данным узлом и всеми потомками этого узла.

Прохождение от родительского узла к его дочернему узлу и к другим потомком осуществляется вдоль ***пути***. Тот факт, что каждый некорневой узел имеет единственного родителя, гарантирует, что существует единственный путь из любого узла к его потомкам. Путь от корня к узлу дает

меру, называемую ***уровнем*** узла. Уровень узла есть длина пути от корня к этому узлу. Уровень корня равен 0. Каждый сын корня является узлом 1 уровня, следующее поколение – 2 уровня и тд.

Глубина дерева есть максимальный уровень любого его узла либо глубина дерева есть длина самого длинного пути от корня до узла.

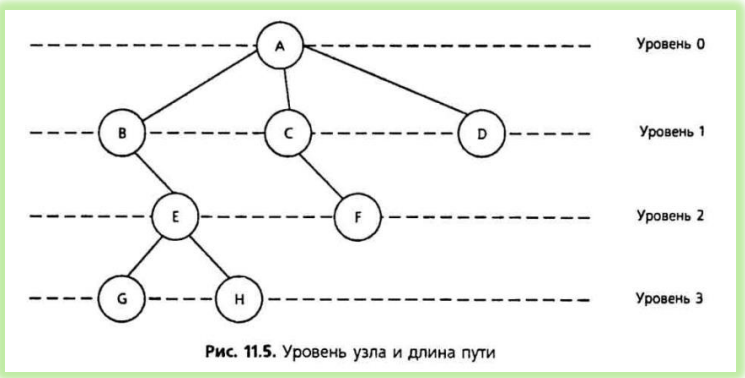
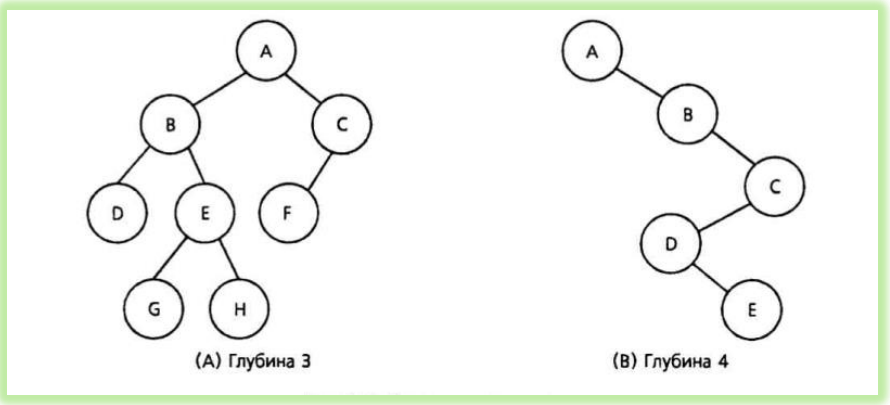


Рис. 11.5. Уровень узла и длина пути

Бинарные деревья



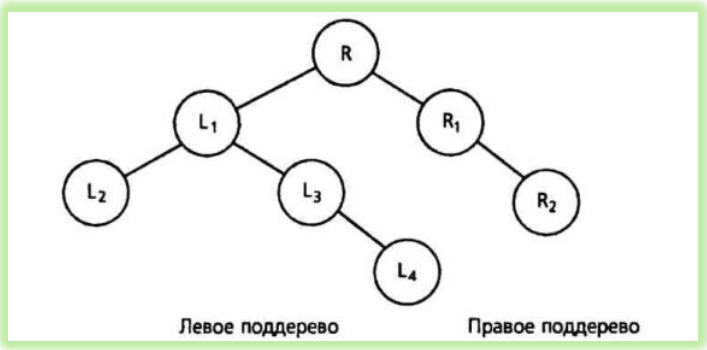
У каждого узла бинарного дерева может быть 0,1 или 2 сына. По отношению к узлу слева будем употреблять термин левый сын, а по отношению к узлу справа – правый сын. Наименования левый и правый относятся к графическому представлению дерева. Бинарное дерево является рекурсивной структурой. Каждый узел – это корень своего собственного поддерева. У него есть сыновья, которые сами являются корнями деревьев, называемых левым и правым поддеревом соответственно.

Бинарное дерево – это такое множество узлов B, что:

- B является деревом, если множество узлов пусто (пустое дерево – тоже дерево)
- B разбивается на три непересекающихся подмножества:

{R} корневой узел
{L_1, L_2, ..., L_m} левое поддерево R
{R_1, R_2, ..., R_m} правое поддерево R

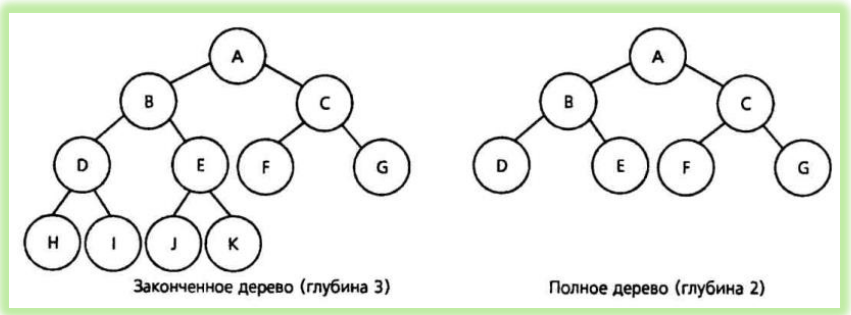
На любом уровне n бинарное дерево может содержать от 1 до 2^n узлов. Число узлов, приходящееся на уровень, является показателем плотности дерева. Интуитивно плотность есть мера величины дерева (число узлов) по отношению к глубине дерева.



Быстрый поиск – главное, что обуславливает использование деревьев для хранения данных.

Вырожденные деревья – такие деревья, у которых есть единственный лист и каждый нелистовой узел имеет только одного сына. Такое дерево эквивалентно связанному списку.

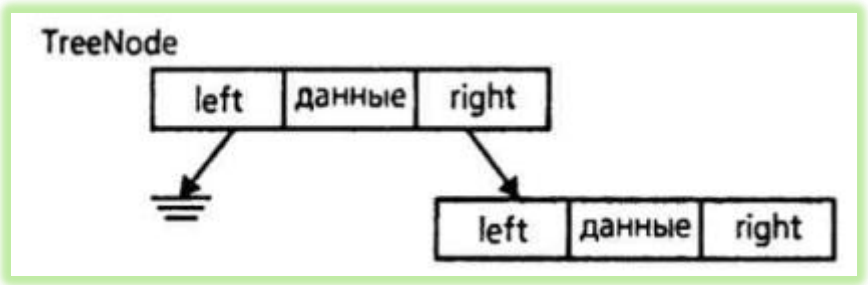
Законченные бинарные деревья - глубины N, где каждый уровень 0..N-1 имеет полный набор узлов и все листья уровня N расположены слева. Законченное бинарное дерево, содержащее 2^N узлов на уровне N является **полным**.



Структура бинарного дерева

Структура бинарного дерева построена из узлов. Как и в связанном списке, эти узлы содержат поля данных и указатели на другие узлы в коллекции.

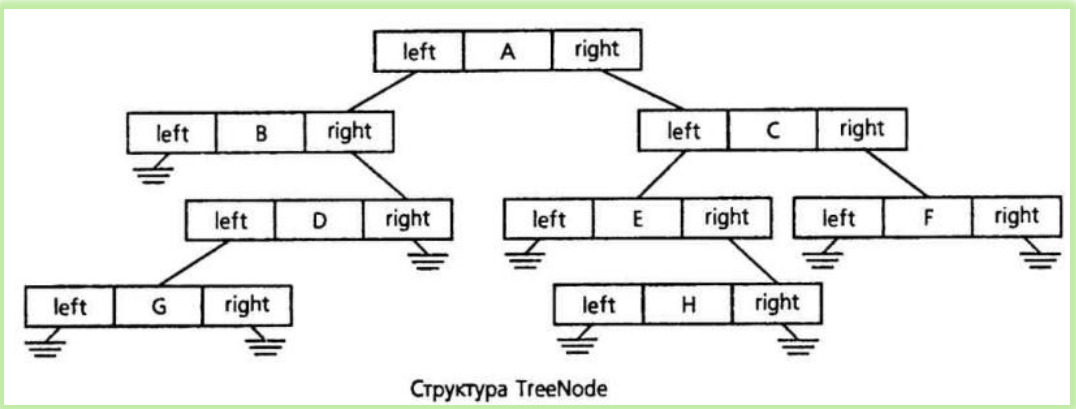
Узел дерева содержит поле данных и два поля с указателями. Поля указателей называются левым и правым указателем, так как они указывают на правое и левое поддерево соответственно. Значение NULL является признаком пустого дерева.



Корневой узел определяет входную точку дерева, а поле указателя – узел следующего уровня. Листовой узел содержит NULL в поле правого и левого указателей.

Проектирование класса TreeNode

В этом разделе разрабатывается класс TreeNode, в котором объявляются объекты-узлы бинарного дерева. Узел состоит из поля данных, которое является открытым (public) элементом, т.е. к которому пользователь может обращаться непосредственно. Это позволяет клиенту читать или обновлять данные во время прохождения дерева, а также допускает возвращение ссылки на данные. Последняя особенность используется более сложными структурами данных, такими как словари. Два поля с указателями являются закрытыми (private) элементами, доступ к которым осуществляется посредством функций Left() и Right(). Объявление и определение класса TreeNode содержатся в файле treenode.h.



Структура TreeNode

Спецификация класса TreeNode

ОБЪЯВЛЕНИЕ

```
// BinSTree зависит от TreeNode
template <class T>
class BinSTree;

// объявление объекта для узла бинарного дерева
template <class T>
class TreeNode
{
  private:
    // указатели левого и правого дочерних узлов
    TreeNode<T> *left;
    TreeNode<T> *right;

  public:
    // открытый элемент, допускающий обновление
    T data;

    // конструктор
    TreeNode (const T& item, TreeNode<T> *lptr = NULL,
                             TreeNode<T> *rptr = NULL);

    // методы доступа к полям указателей
    TreeNode<T>* Left(void) const;
    TreeNode<T>* Right(void) const;

    // сделать класс BinSTree дружественным, поскольку необходим
    // доступ к полям left и right
    friend class BinSTree<T>;
};
```

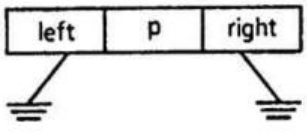
ОПИСАНИЕ

Построение бинарного дерева

Бинарное дерево состоит из коллекции объектов TreeNode, связанных посредством своих полей с указателями. Объект TreeNode создается динамически с помощью функции new.

```
TreeNode<int> *p;      // объявление указателя
                       // на целочисленный узел дерева

p = new TreeNode(item); // левый и правый указатели равны NULL
```



Вызов функции new обязательно должен включать значение данных. Если в качестве параметра передается также указатель объекта TreeNode, то он используется вновь созданным узлом для присоединения дочернего узла. Определим функцию GetTreeNode, принимающую данные и ноль или более указателей объекта TreeNode для создания и инициализации узла бинарного

дерева. При недостаточном количестве доступной памяти программа прекращается сразу после выдачи сообщения об ошибке.

```
// создать объект TreeNode с указательными полями lptr и rptr.
// по умолчанию указатели содержат NULL.
template <class T>
TreeNode<T> *GetTreeNode(T item, TreeNode<T> *lptr = NULL,
                         TreeNode<T> *rptr = NULL)
{
  TreeNode<T> *p;

  // вызвать new для создания нового узла
  // передать туда параметры lptr и rptr
  p = new TreeNode<T> (item, lptr, rptr);

  // если памяти недостаточно, завершить программу сообщением об ошибке
  if (p == NULL)
  {
    cerr << "Ошибка при выделении памяти!\n";
    exit(1);
  }

  // вернуть указатель на выделенную системой память
  return p;
}
```

Функция FreeTreeNode принимает указатель на объект TreeNode и освобождает занимаемую узлом память, вызывая функцию c++ delete.

```
// освободить динамическую память, занимаемую данным узлом
template <class t>
void FreeTreeNode(TreeNode<T> *p)
{
  delete p;
}
```

Обе эти функции находятся в файле treelib.h вместе с функциями обработки бинарного дерева.

Источники:

- «Структуры данных в C++» - Уильям Топ Уильям Форд
- https://ru.wikipedia.org/wiki/Дерево_(структура_данных)