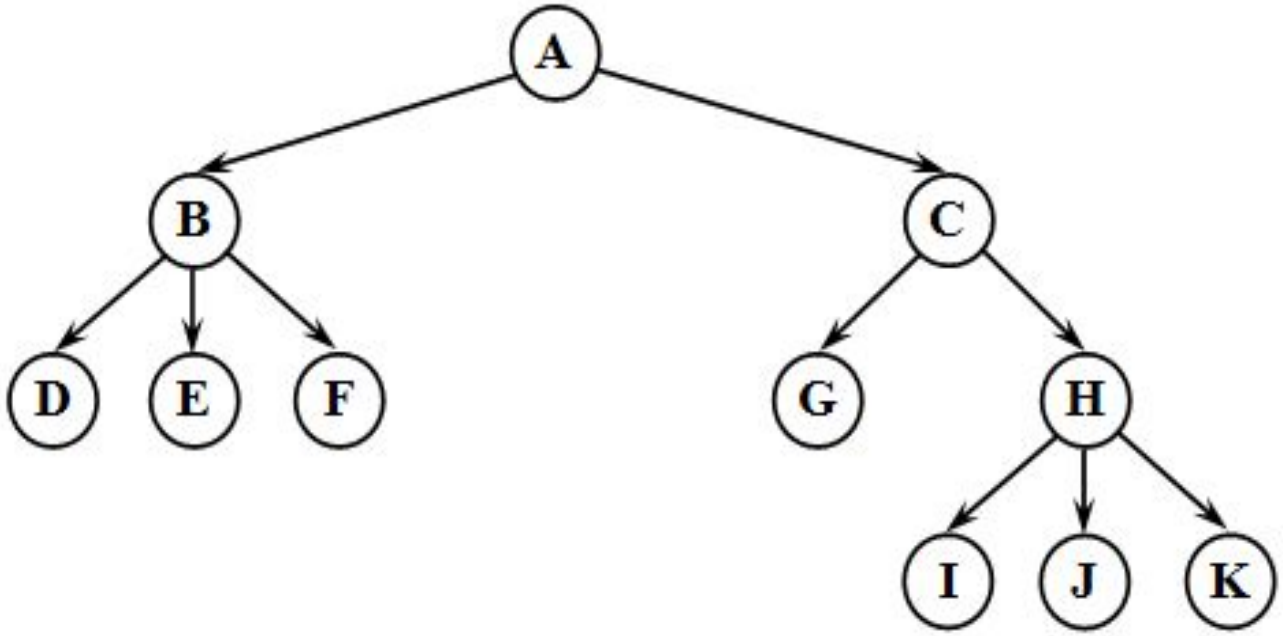
Лекция 4. Динамические структуры данных: Деревья

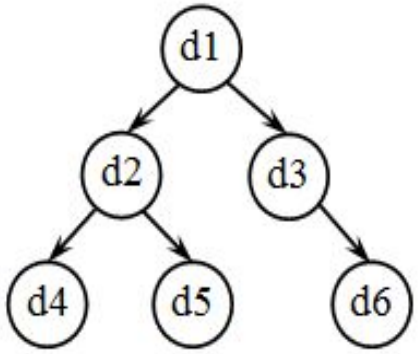
1. Основные определения
2. Бинарные деревья
3. Бинарная куча
4. N-арные деревья (B-деревья)

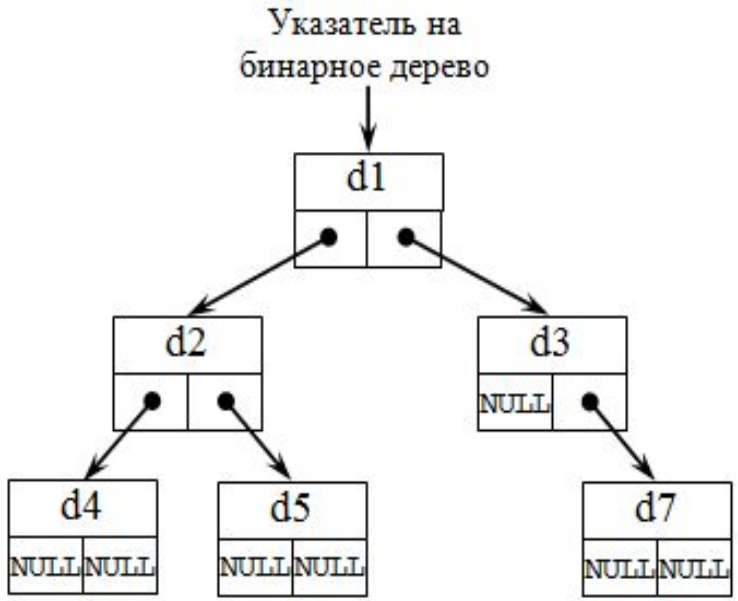
Основные определения

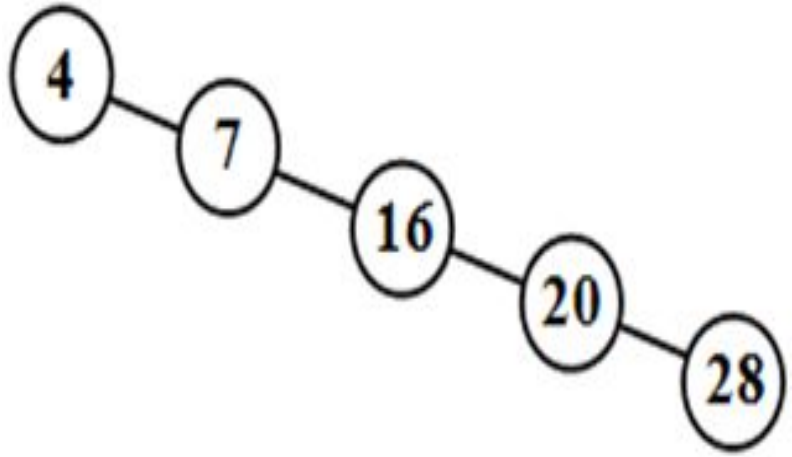


Дерево – структура данных, представляющая собой древовидную структуру в виде набора связанных узлов.

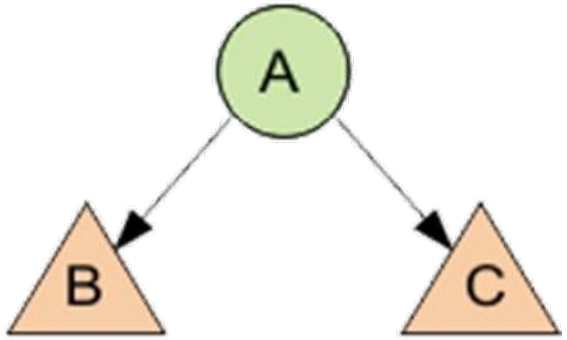
Бинарное дерево



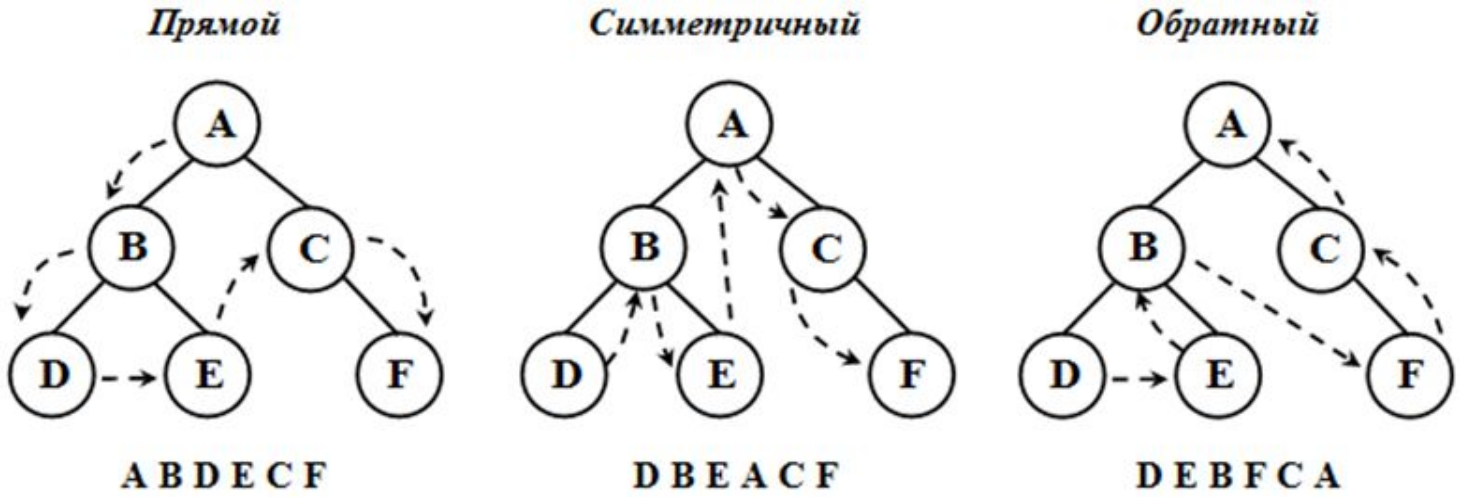




Обход бинарного дерева



- прямой; (префиксный) (ABC)
- симметричный;(BAC) (Инфиксный)
- обратный (постфиксный (BCA)



Реализация бинарного дерева

```
struct tnode {
  int field;          // поле данных
  struct tnode *left;  // левый потомок
  struct tnode *right; //правый потомок
};
```

Добавление узлов в дерево

```
struct tnode * addnode(int x, tnode *tree) {
  if (tree == NULL) { // Если дерева нет, то формируем корень
    tree =new tnode; // выделяем память под узел
    tree->field = x;   // поле данных
    tree->left =  NULL;
    tree->right = NULL; // ветви инициализируем пустотой
  }
 else  if (x < tree->field)   // условие добавления левого потомка
    tree->left = addnode(x,tree->left);
  else    // условие добавления правого потомка
    tree->right = addnode(x,tree->right);
  return(tree);
}
```

Реализация прямого обхода дерева

```
void treeprint(tnode *tree) {
  if (tree!=NULL) { //Пока не встретится пустой узел
    cout << tree->field; //Отображаем корень дерева
    treeprint(tree->left); //Рекурсивная функция для левого поддерева
    treeprint(tree->right); //Рекурсивная функция для правого поддерева
  }
}
```

Реализация симметричного обхода дерева

```
void treeprint(tnode *tree) {

  if (tree!=NULL) { //Пока не встретится пустой узел

      treeprint(tree->left); //Рекурсивная функция для левого поддерева

    cout << tree->field; //Отображаем корень дерева

    treeprint(tree->right); //Рекурсивная функция для правого поддерева

  }

}
```

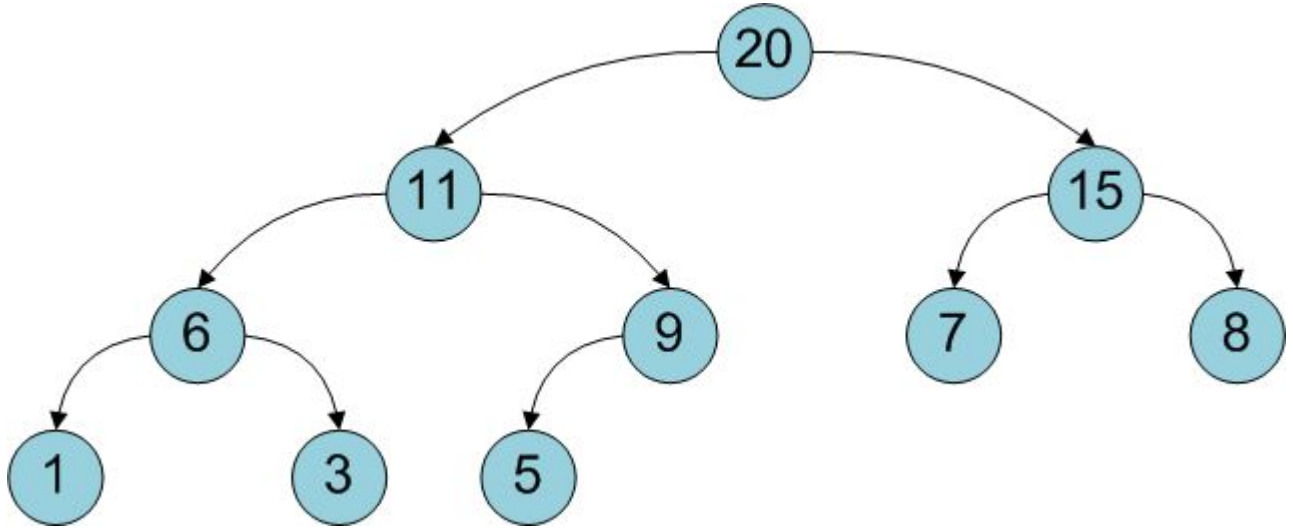
Реализация обратного обхода дерева

```
void treeprint(tnode *tree) {
  if (tree!=NULL) { //Пока не встретится пустой узел
    treeprint(tree->left); //Рекурсивная функция для левого поддерева
    treeprint(tree->right); //Рекурсивная функция для правого поддерева
    cout << tree->field; //Отображаем корень дерева
  }
}
```

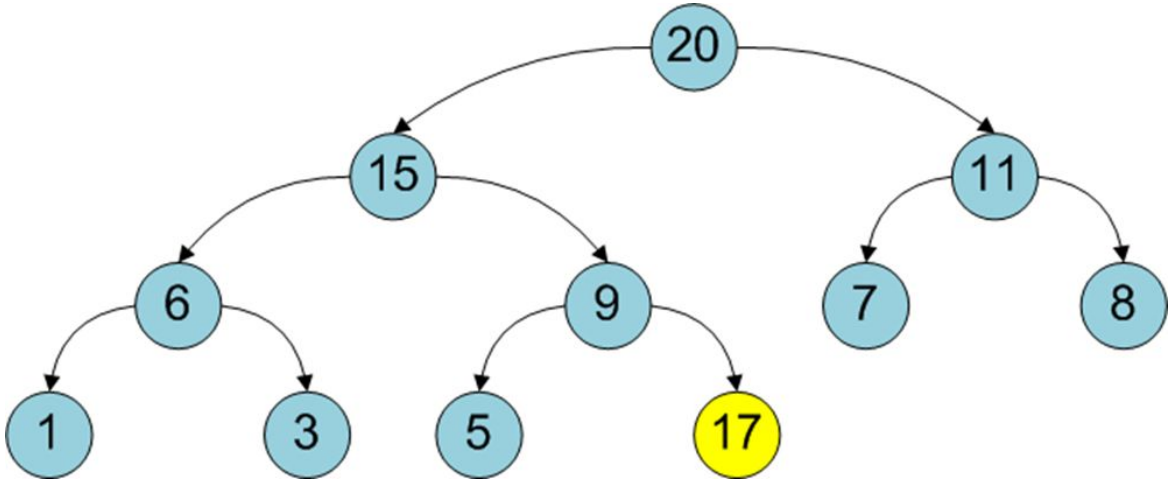
Бинарная куча

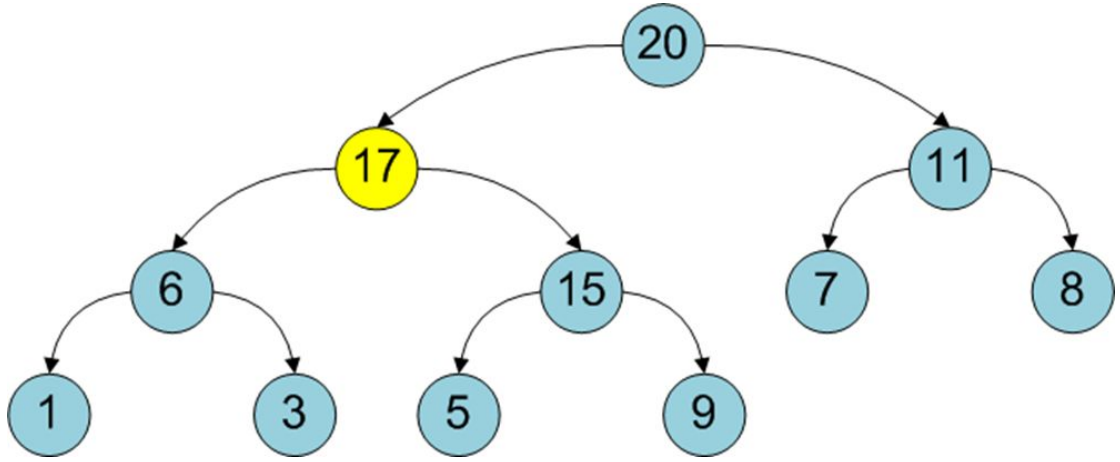
Бинарная куча - это бинарное дерево, для которого выполняется *основное свойство кучи*: приоритет (значение) каждой вершины больше приоритет

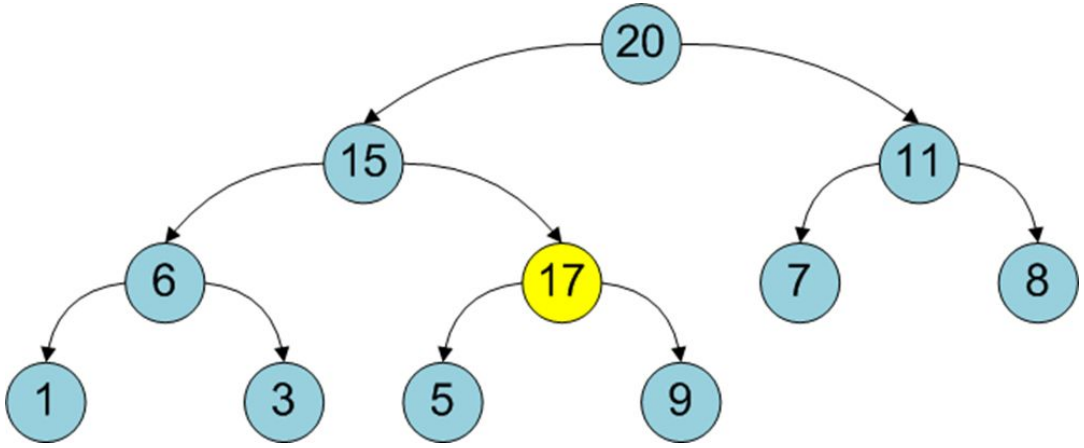
ов(значений) её потомков.



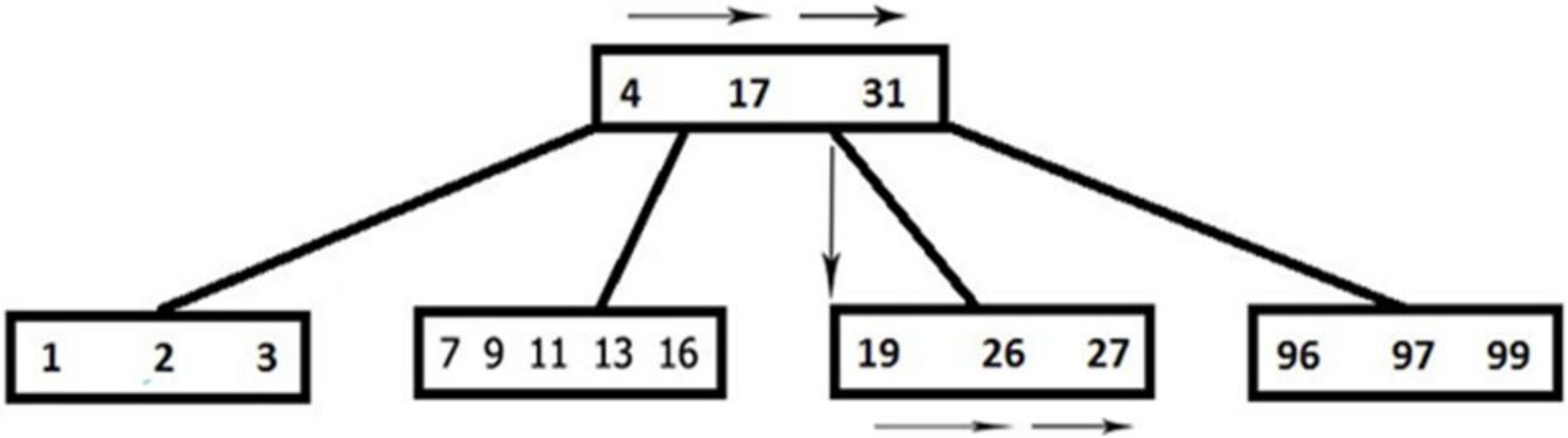
Добавление элемента в кучу







N-арное дерево (B-дерево)



Высота B-дерева с $n \geq 1$ узлами и минимальной степенью $t \geq 2$ не превышает $\log_t(n+1)$.