

## Задание 3

Деменчук Георгий ПИ19-4

- Выполнить прямой обход (в ручном режиме и программную реализацию) бинарного дерева, представленного на рисунке 2.
- Выполнить обратный обход (в ручном режиме и программную реализацию) бинарного дерева, представленного на рисунке 2.
- Выполнить симметричный обход (в ручном режиме и программную реализацию) бинарного дерева, представленного на рисунке 2.

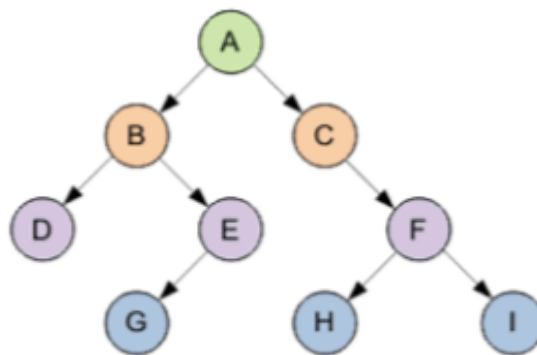


Рисунок 2

## Основная программная реализация

In [1]:

```
from tree_module import BinaryTree
```

In [2]:

```
class Orders:
    """Стат методы для обхода дерева"""

    @staticmethod
    def preorder(tree):
        """Прямой обход дерева"""
        if tree:
            print(tree.get_root_val())
            Orders.preorder(tree.get_left_child())
            Orders.preorder(tree.get_right_child())

    @staticmethod
    def inorder(tree):
        """Симметричный обход дерева"""
        if tree != None:
            Orders.inorder(tree.get_left_child())
            print(tree.get_root_val())
            Orders.inorder(tree.get_right_child())

    @staticmethod
    def postorder(tree):
        """Обратный обход"""
        if tree != None:
            Orders.postorder(tree.get_left_child())
            Orders.postorder(tree.get_right_child())
            print(tree.get_root_val())
```

In [3]:

```
# Элемент на 1 уровне
A = BinaryTree('A')

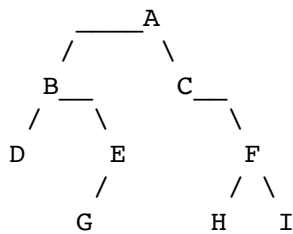
#Элементы на 2 уровне
B = A.insert_left('B')
C = A.insert_right('C')

#Элементы на 3 уровне
D = B.insert_left("D")
E = B.insert_right("E")
F = C.insert_right("F")

#Элементы на 4 уровне
E.insert_left("G")
F.insert_left("H")
F.insert_right("I")

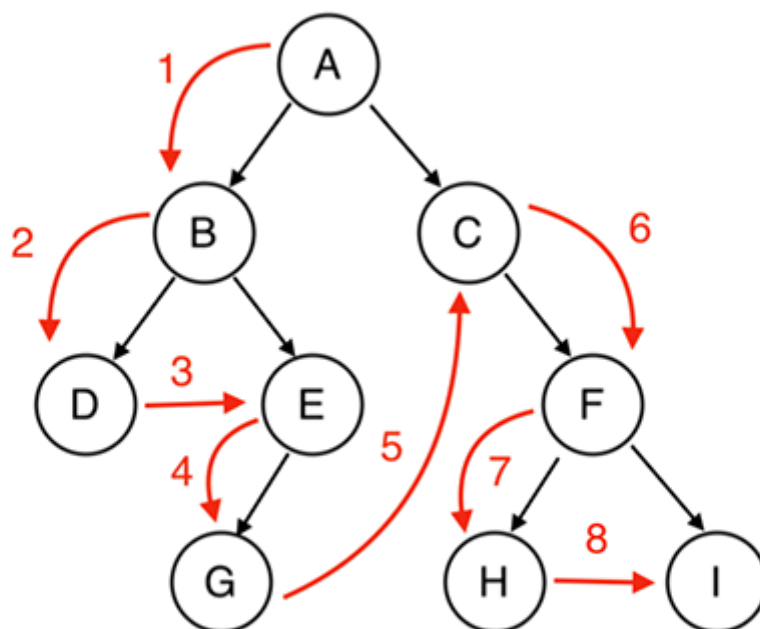
print("Построенное дерево:")
print(A)
```

Построенное дерево:



## Прямой обход

### Ручной режим



Обход: A B D E G C F H I

## Программная реализация

In [4]:

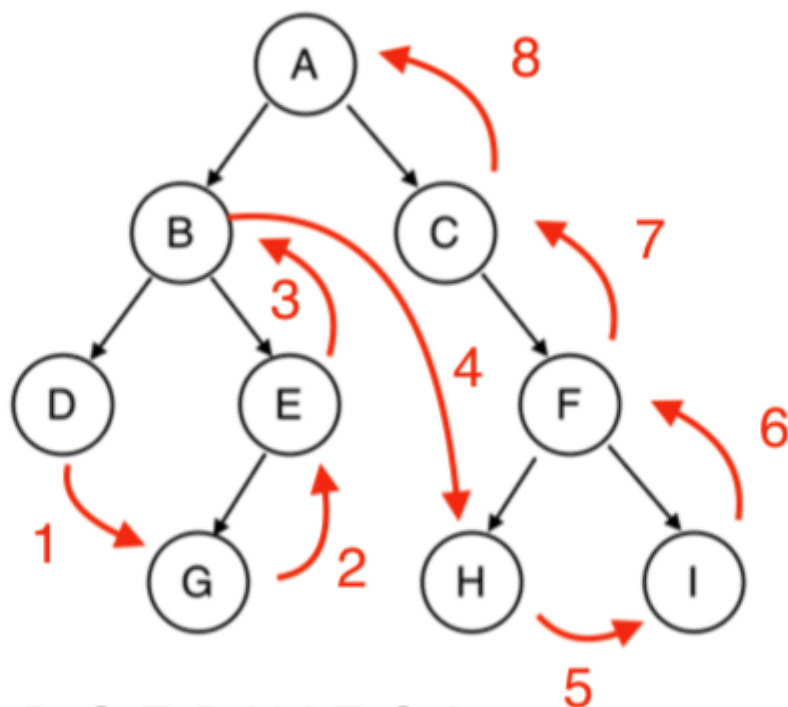
```
#Обход в прямом порядке  
print("Прямой обход дерева:")  
Orders.preorder(A)
```

Прямой обход дерева:

A  
B  
D  
E  
G  
C  
F  
H  
I

## Обратный обход

### Ручной режим



Обход: D G E B H I F C A

## Программная реализация

In [5]:

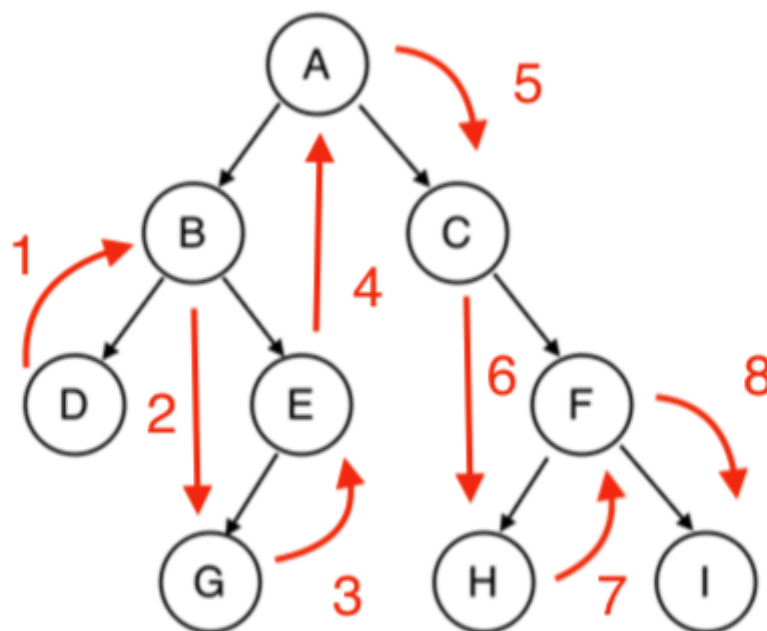
```
#Обратный обход
print("Обратный обход дерева:")
Orders.postorder(A)
```

Обратный обход дерева:

D  
G  
E  
B  
H  
I  
F  
C  
A

## Симметричный обход

### Ручной режим



Обход: D B G E A C H F I

## Программная реализация

In [6]:

```
#Симметричный обход
print("Симметричный обход дерева:")
Orders.inorder(A)
```

Симметричный обход дерева:

D  
B  
G  
E  
A  
C  
H  
F  
I

In [ ]: