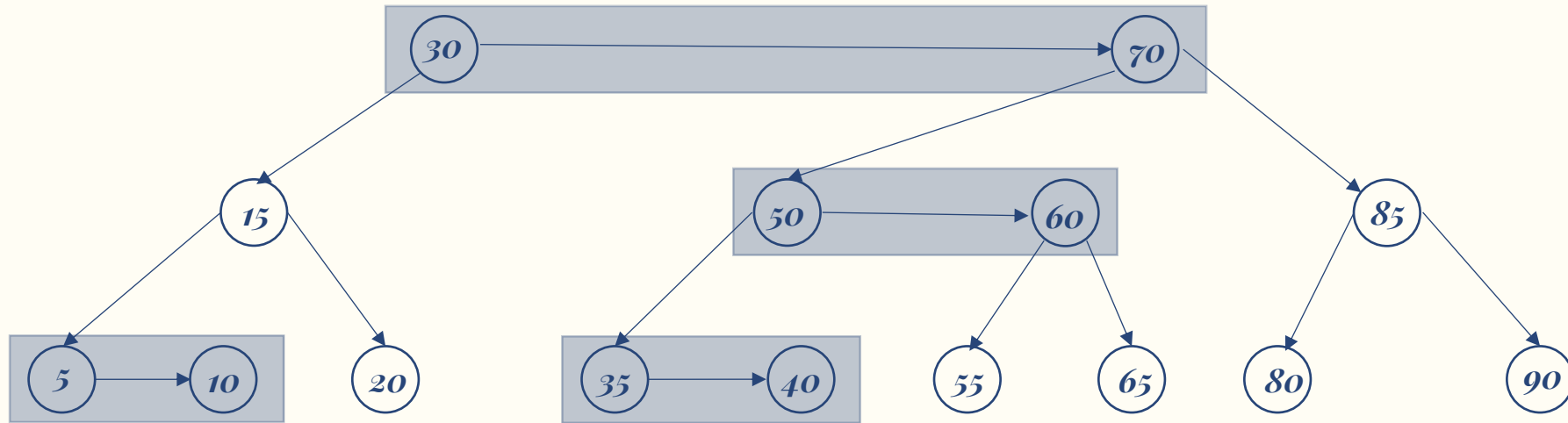


[двоичное дерево поиска]

# *АА-дерево*

Попова Анастасия 11-401



**AA-дерево** — структура данных, представляющая собой *сбалансированное двоичное дерево поиска*, которое является разновидностью *красно-черного дерева* с дополнительными ограничениями. Служит для эффективного хранения и поиска данных с автоматической балансировкой.

# *Особенности*

В отличие от красно-черных деревьев, к одной вершине можно присоединить вершину только того же уровня, только одну и только справа (другими словами, красные вершины могут быть добавлены только в качестве правого ребенка).

## *Свойства*

- Уровень каждого листа равен 1.
- Уровень каждого левого ребенка ровно на один меньше, чем у его родителя.
- Уровень каждого правого ребенка равен или на один меньше, чем у его родителя.
- Уровень каждого правого внука строго меньше, чем у его прародителя.
- Каждая вершина с уровнем больше 1 имеет двоих детей.

# *Балансировка*

*skew*

*split*

— *устранение левого горизонтального ребра.*  
Делаем правое вращение, чтобы заменить поддерево, содержащее левую горизонтальную связь, на поддерево, содержащее разрешенную правую горизонтальную связь.

— *устранение двух последовательных правых горизонтальных ребер.* Делаем левое вращение и увеличиваем уровень, чтобы заменить поддерево, содержащее две или более последовательных правильных горизонтальных связи, на вершину, содержащую два поддерева с меньшим уровнем.

# *Плюсы АА-дерева*

- 1 Автоматическая балансировка* после вставки или удаления элемента
- 2 Проще реализовать*, чем красно-черное дерево
- 3 Эффективные операции* вставки, удаления и поиска

# *Минусы АА-деревя*

- 1* *Балансировка строже, чем у красно-черных деревьев*
- 2* *Использование дополнительной памяти на хранение уровня каждого узла*

# *Где лучше применять?*

*АА-дерево можно использовать для создания и ведения упорядоченных списков, где нам будет нужно больше искать значения, а не добавлять или удалять их:*

*Базы данных:* реализация упорядоченных ассоциативных массивов (например, Java's TreeMap).

*Алгоритмы планирования задач:* в Linux's Completely Fair Scheduler для управления приоритетами процессов.

*Хеш-таблицы:* оптимизация работы с большими объемами данных.

*Кэширование:* хранение часто используемых данных с быстрой доступностью.