BRANCH AND BOUND

- A algorithmic technique or problem solving strategy similar to backtracking
- Uses State Space Tree for solving a problem. Solution is represented in the form of a state space tree.
- It is used for solving Optimization Problems – "Minimization Problems" only.
- Provides only an OPTIMAL solution to a problem
- Uses the Breadth-first search method for exploring solutions in a state space tree.
- We don't explore all the nodes in a branch and bound algorithm.
- Bounding Function:
  - kill some live nodes without actually expanding them.
  - Heuristic function used to calculate the lower and upper bound values on the possible solutions at each node.

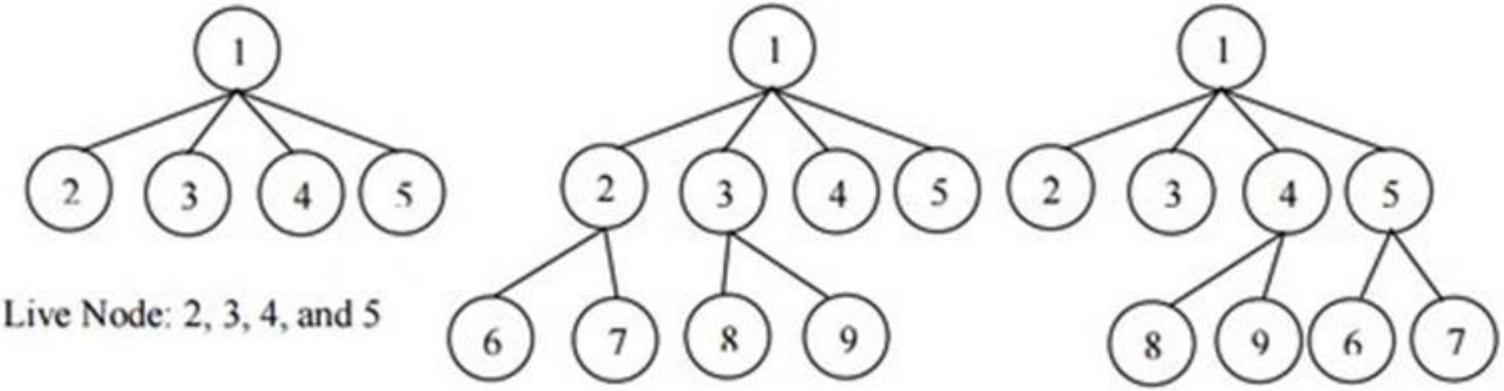
Applications

- Travelling salesman problem
- 0/1 knapsack problem
- Maximum Satisfiability problem (MAX-SAT)
- Flow shop scheduling

Types of BB

- 3 types based on the order in which the state space tree is searched.
- **FIFO Branch and Bound**
  - It maintains the list of live nodes in first-in-first-out order i.e, in a **queue**
  - The live nodes are searched in the FIFO order to make them next E-nodes.
- **LIFO Branch and Bound**
  - It maintains the list of live nodes in last-in-first-out order i.e. in a **stack**.
  - The live nodes are searched in the LIFO order to make them next E-nodes.
- **LC-Branch and Bound – LC -> Least Cost**
  - It uses a **heuristic cost function** to compute the bound values at each node.
  - Nodes are added to the list of live nodes as soon as they get generated.
  - The node with the least value of a cost function selected as a next E-node.

- Example: FIFO-BB & LIFO-BB

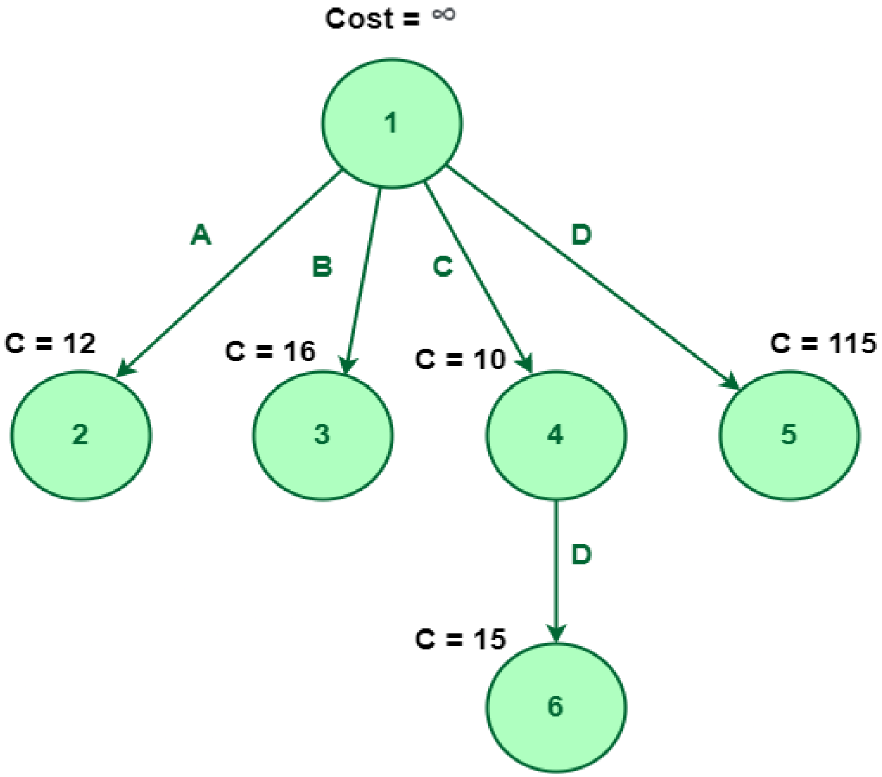


Live Node: 2, 3, 4, and 5

FIFO Branch & Bound
Children of E-node are
inserted in a queue.

LIFO Branch & Bound
Children of E-node are inserted in a
stack.

- Example: LC-BB


Cost = ∞
1
A
B
C
D
C = 12
C = 16
C = 10
C = 115
2
3
4
5
D
C = 15
6