

V(10) => [15,12,25]

V(25) => [10,101]

V(101) => [25,12]

V(12) => [10,101,35]

V(35) => [12]

V(15) => [10]

start : 10

stack(v(10)) : 1

v(10) => stack ([15,12,25]) = 3 : visited : [v(10)]

v(25) => stack ([15,12,10,101]) = 2: visited : [v(10),v(25)]

v(101) => stack ([15,12,10,25,12]) = 2 visited : [v(10),v(25), v(101)]

v(12) => stack ([15,12,10,25,10,101,35]) = 3

visited : [v(10),v(25), v(101),v(12)]

V(35) => stack ([15,12,10,25,10,101,12]) = 1

visited : [v(10),v(25), v(101),v(12),v(35)]

v(12) = 0 stack ([15,12,10,25,10,101]

v(101) = 0 stack ([15,12,10,25,10])

v(10) = 0 stack ([15,12,10,25])

v(25) = 0 stack ([15,12,10])

v(10) = 0 stack ([15,12])

v(12) = 0 stack ([15])

v(15) = 1 stack ([10]) visited : [v(10),v(25), v(101),v(12),v(35),v(15)]

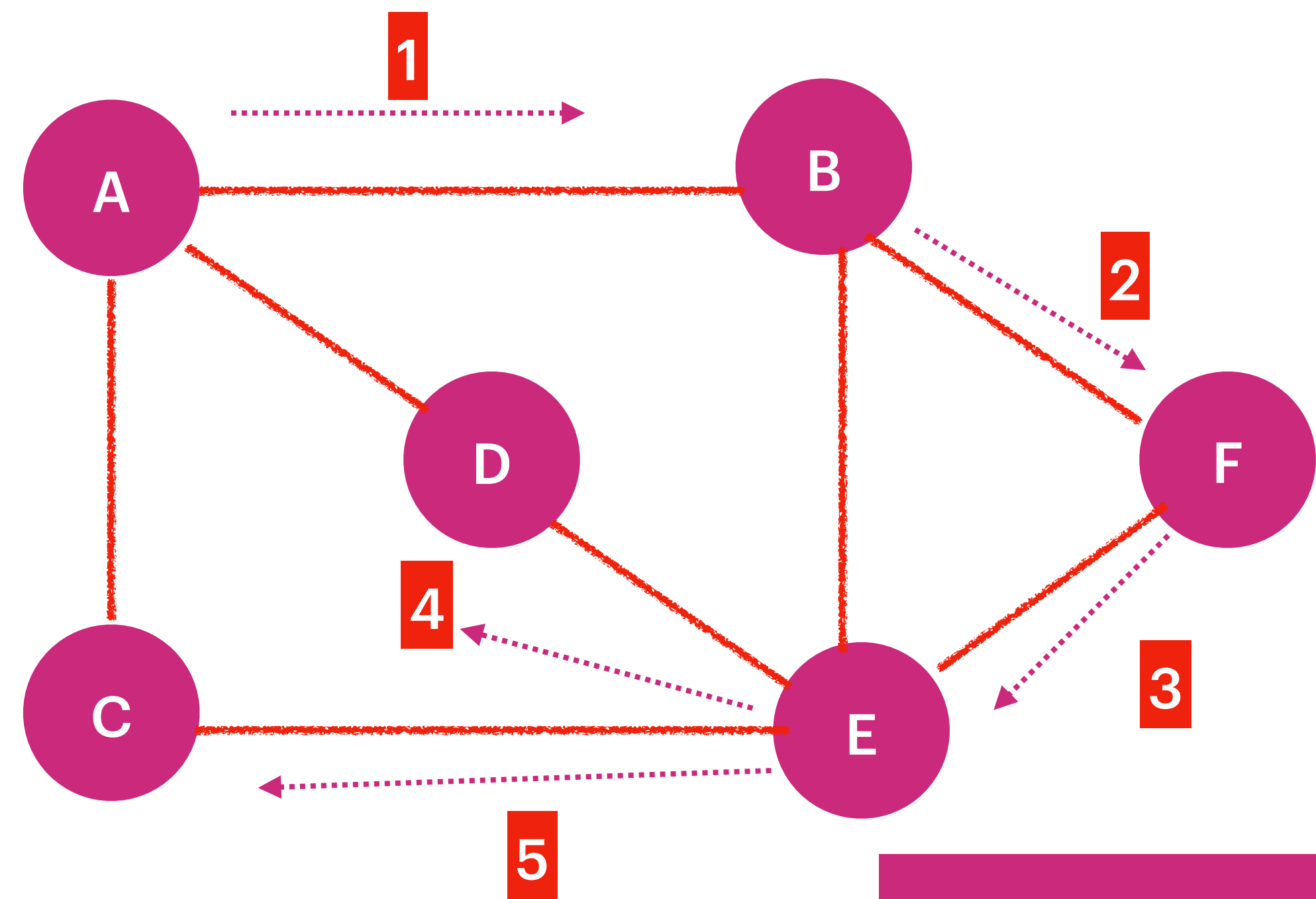
v(10) = 0 stack ([10]) visited : [v(10),v(25), v(101),v(12),v(35),v(15)]

numberOf.Connections(V) + Unique[Edges]

$$13V + 12(E) = V + E$$

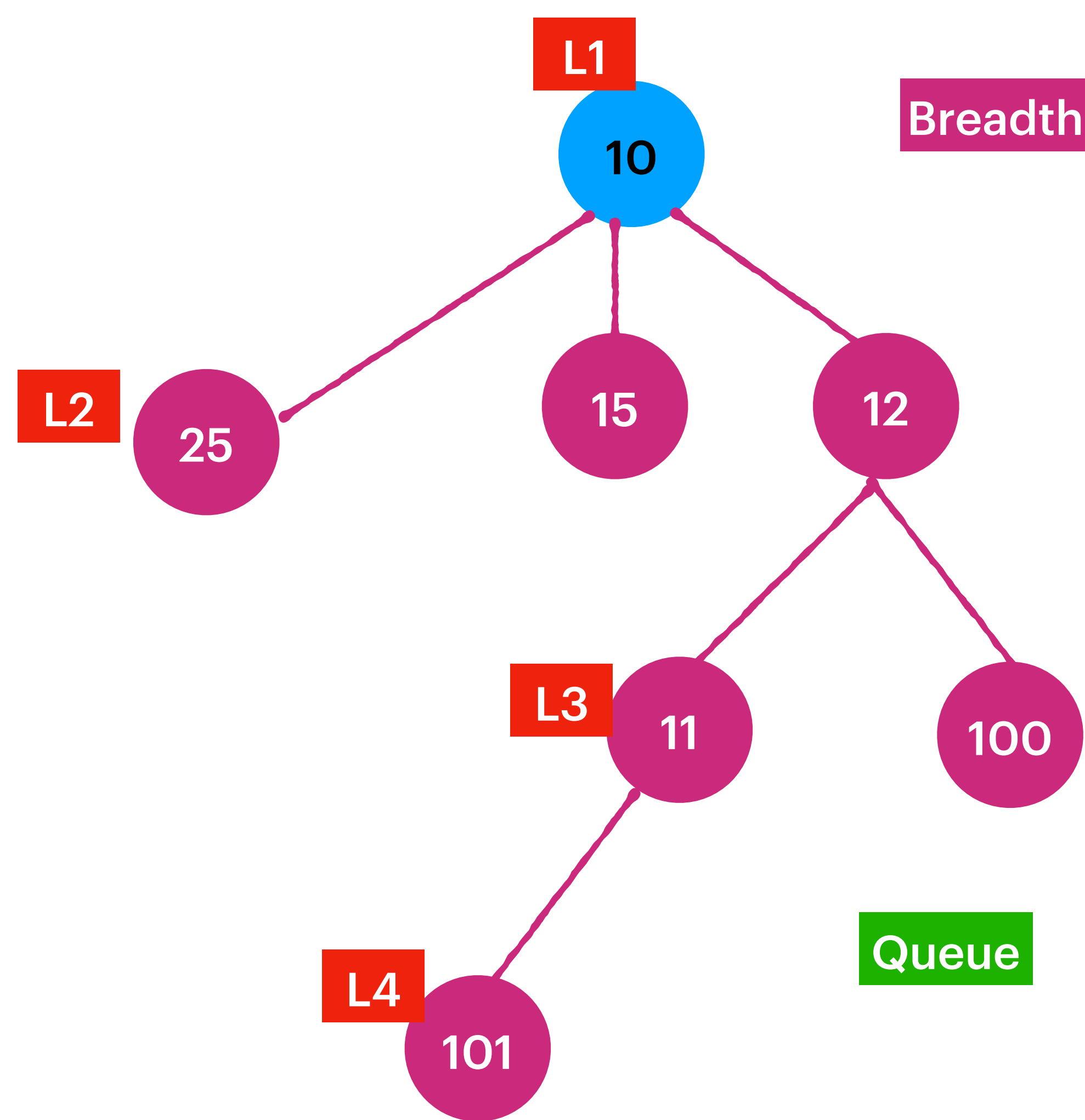
Analysis's Of DFS Time Complexity : $O(V+E)$

Depth First Search



A = [C,D,B]
B = [A,E,F]
C = [A,E]
D = [A,E]
E = [C,D,B,F]
F =[E,B]

Stack []
Visited [A,B,F,E,D,C]
DFS [A,B,F,E,D,C]



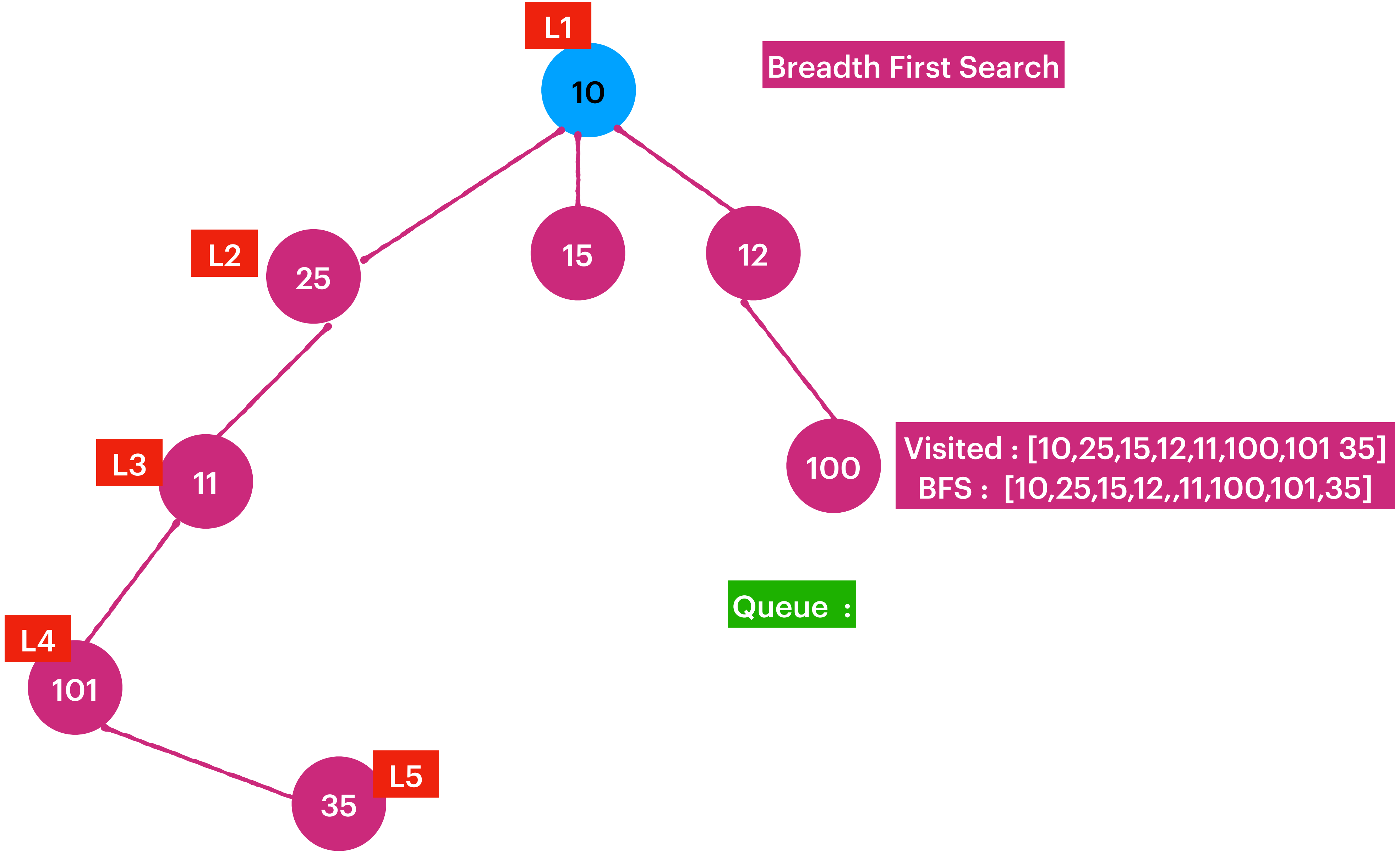
Breadth First Search

10 = [25,15,12]
25 = [10]
15 = [10]
12 = [10,11,100]
11 = [12,101]
100 = [12]
101 = [11]

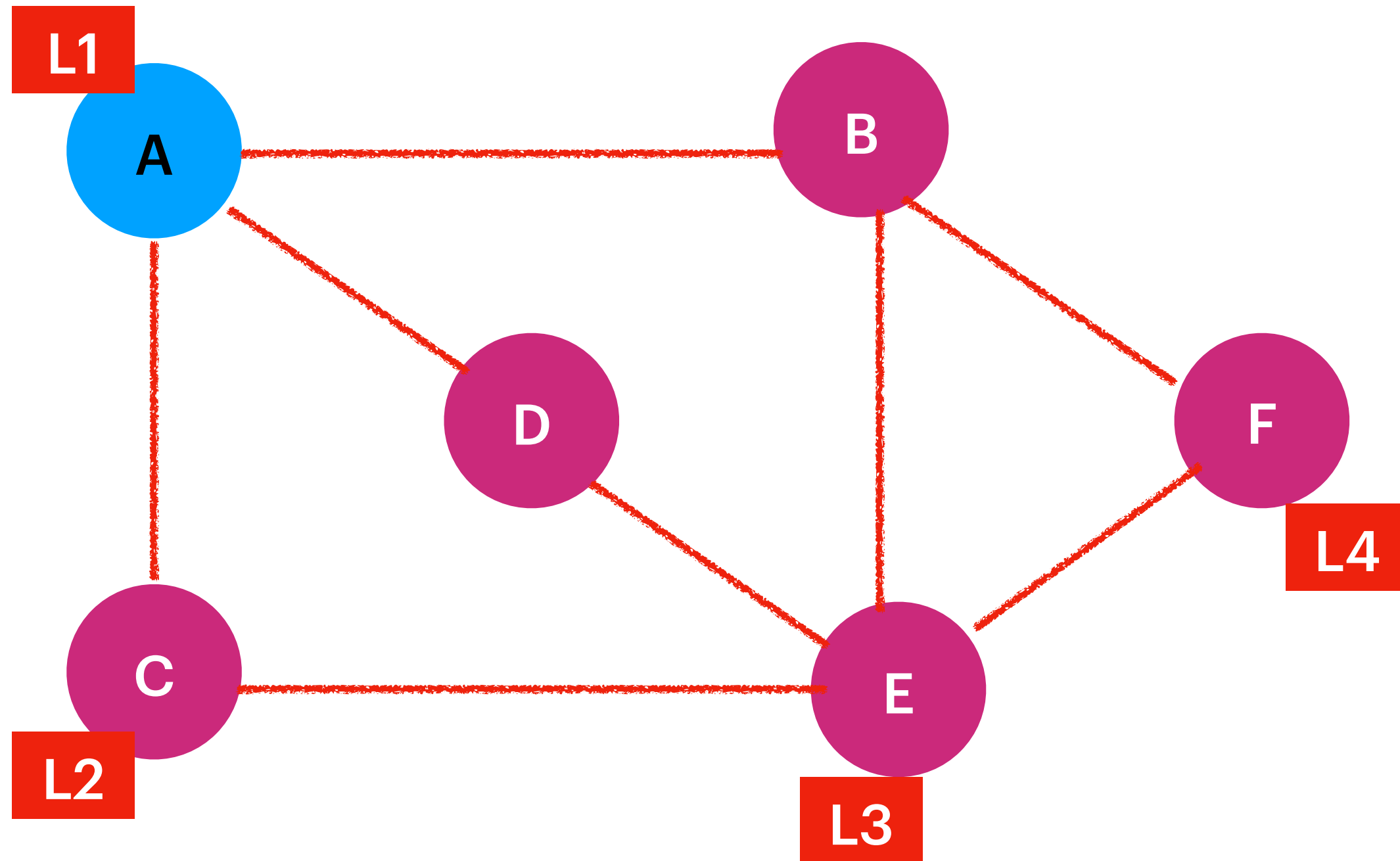
Visited : [10,25,15,12,11,100,101]

BFS : [10,25,15,12,11,100,101]

BFS [10, 25, 15, 12, 11, 100, 101]



Breadth First Search



A = [C,D,B]
B = [A,E,F]
C = [A,E]
D = [A,E]
E = [C,D,B,F]
F = [E,B]

BFS : A,C,D,B,E,F