Problem Reduction Search

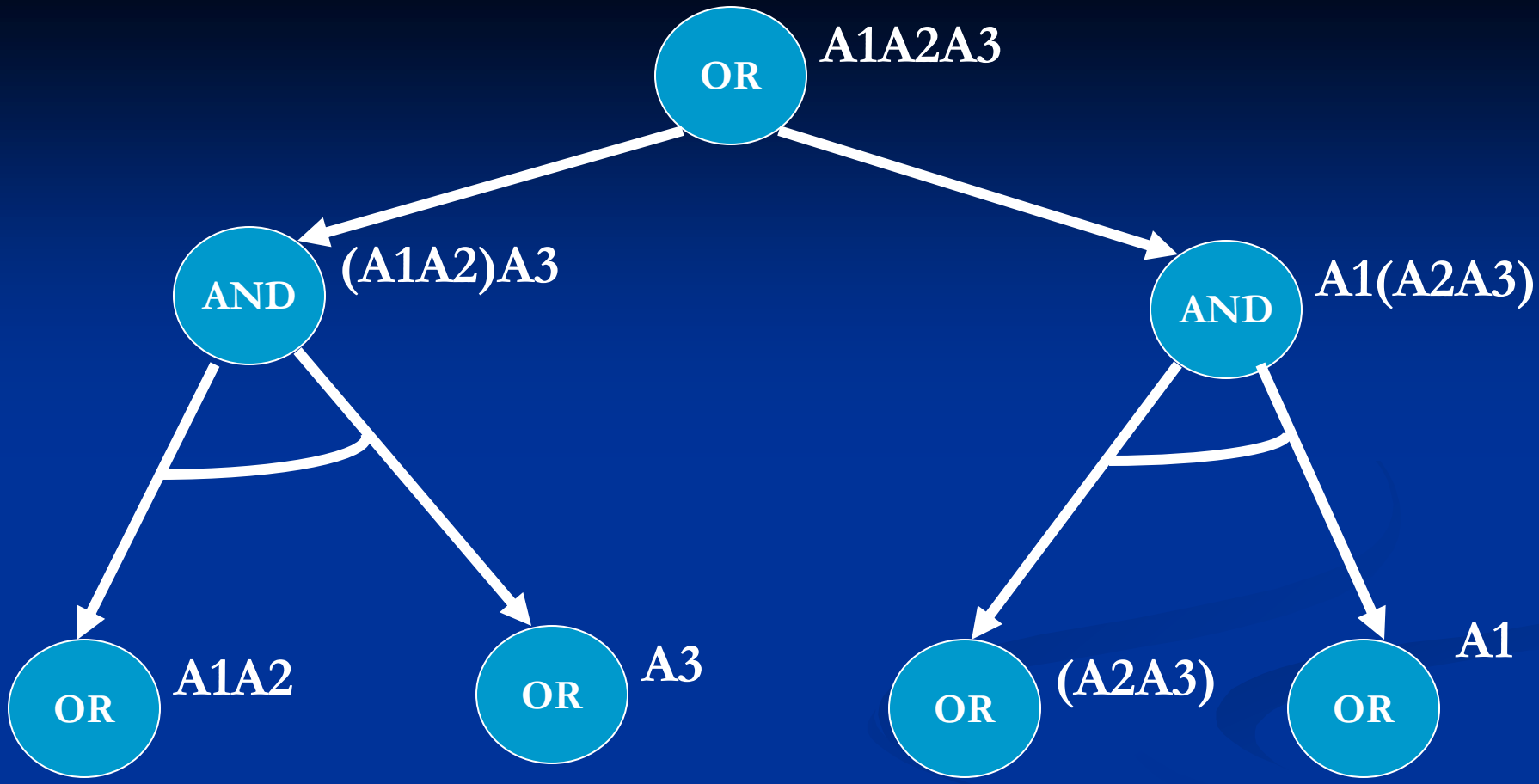
AND/OR Graphs (AO* Algorithm)

Problem Reduction Search

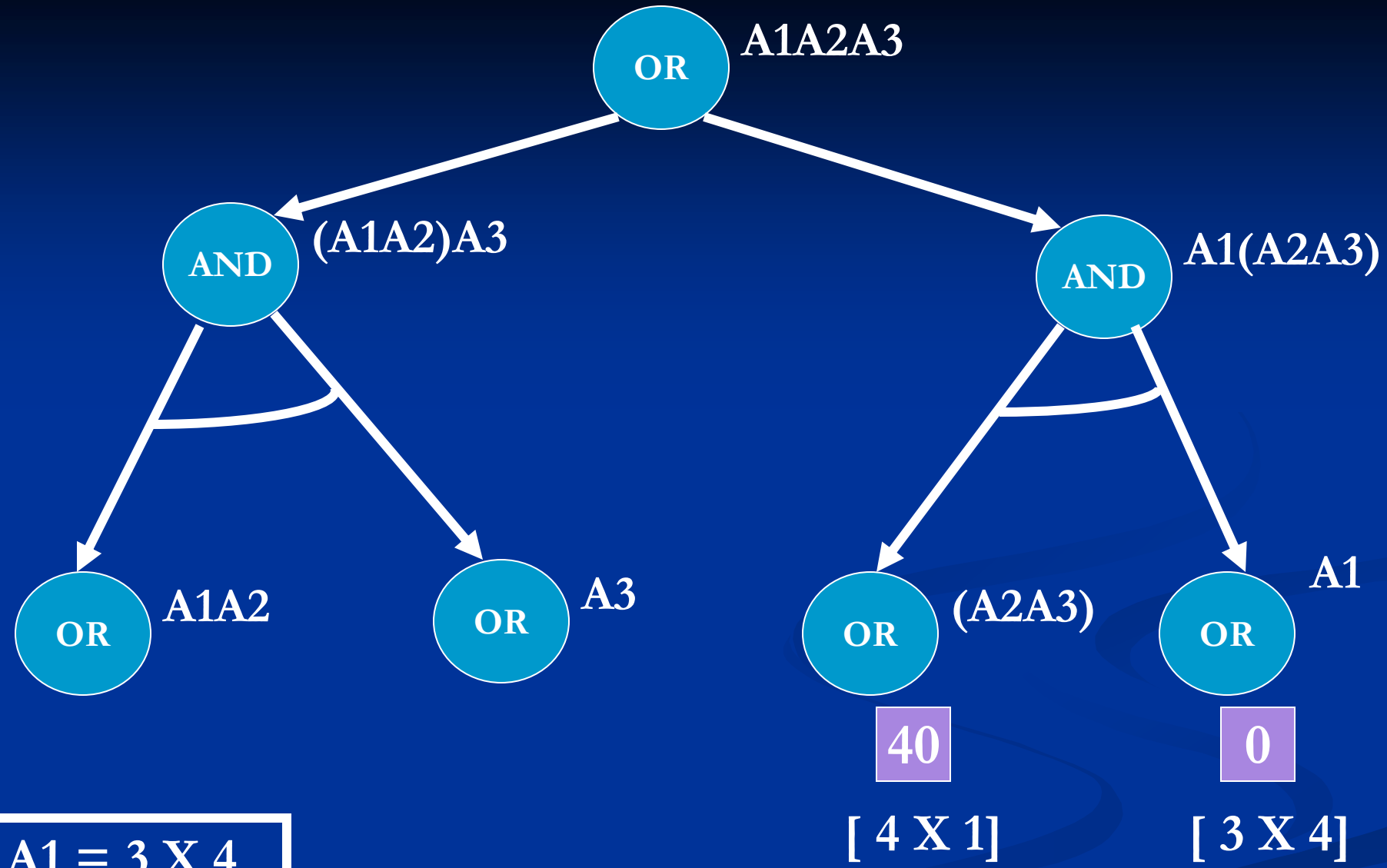
- Planning how best to solve a problem that can be recursively decomposed into sub problems in multiple ways.

Formulations

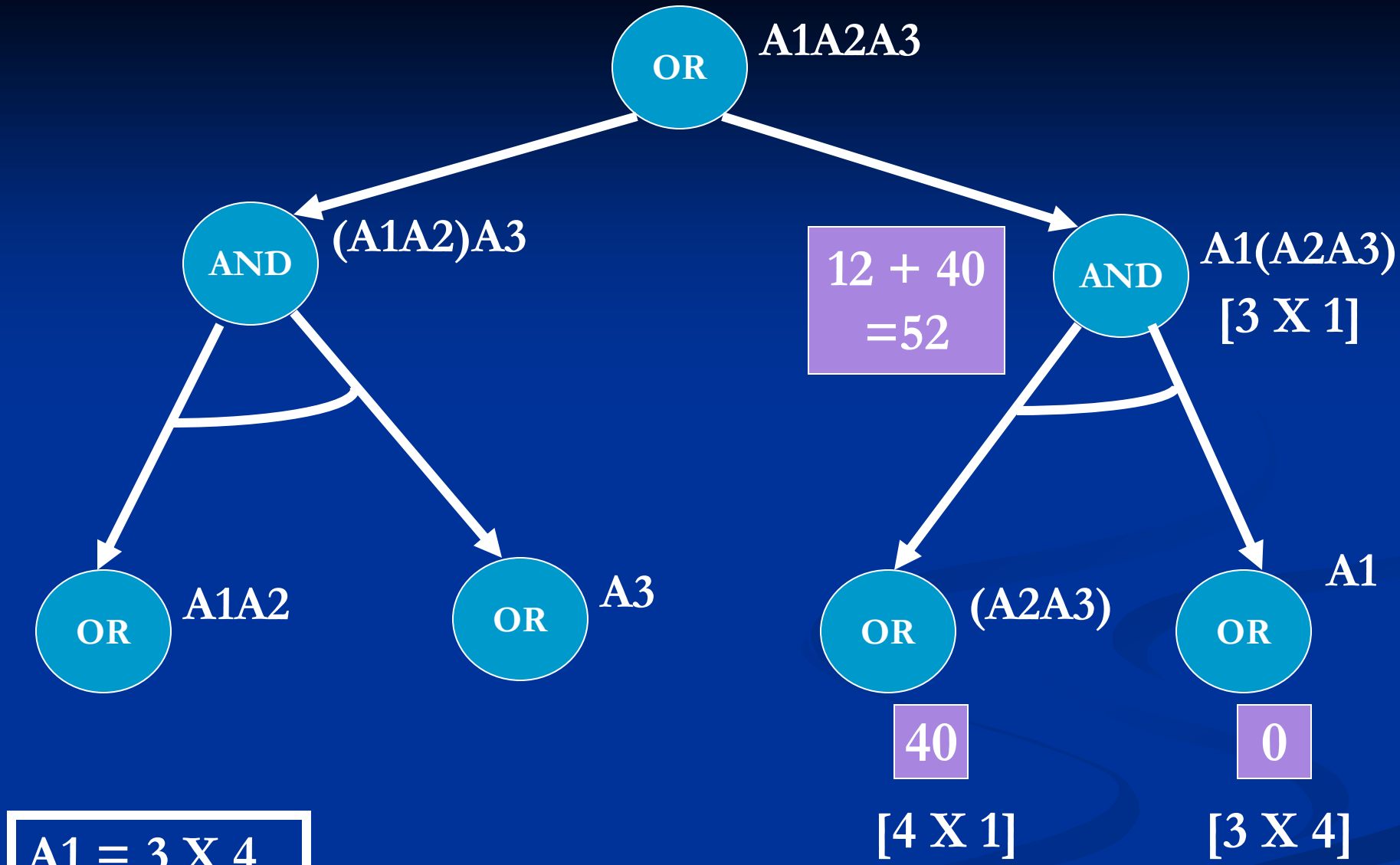
- **AND/OR Graphs**
  - An OR node represents a choice between possible decompositions
  - An AND node represents a given decomposition
- **Game Trees**
  - Max nodes represent the choice of my opponent
  - Min nodes represent my choice



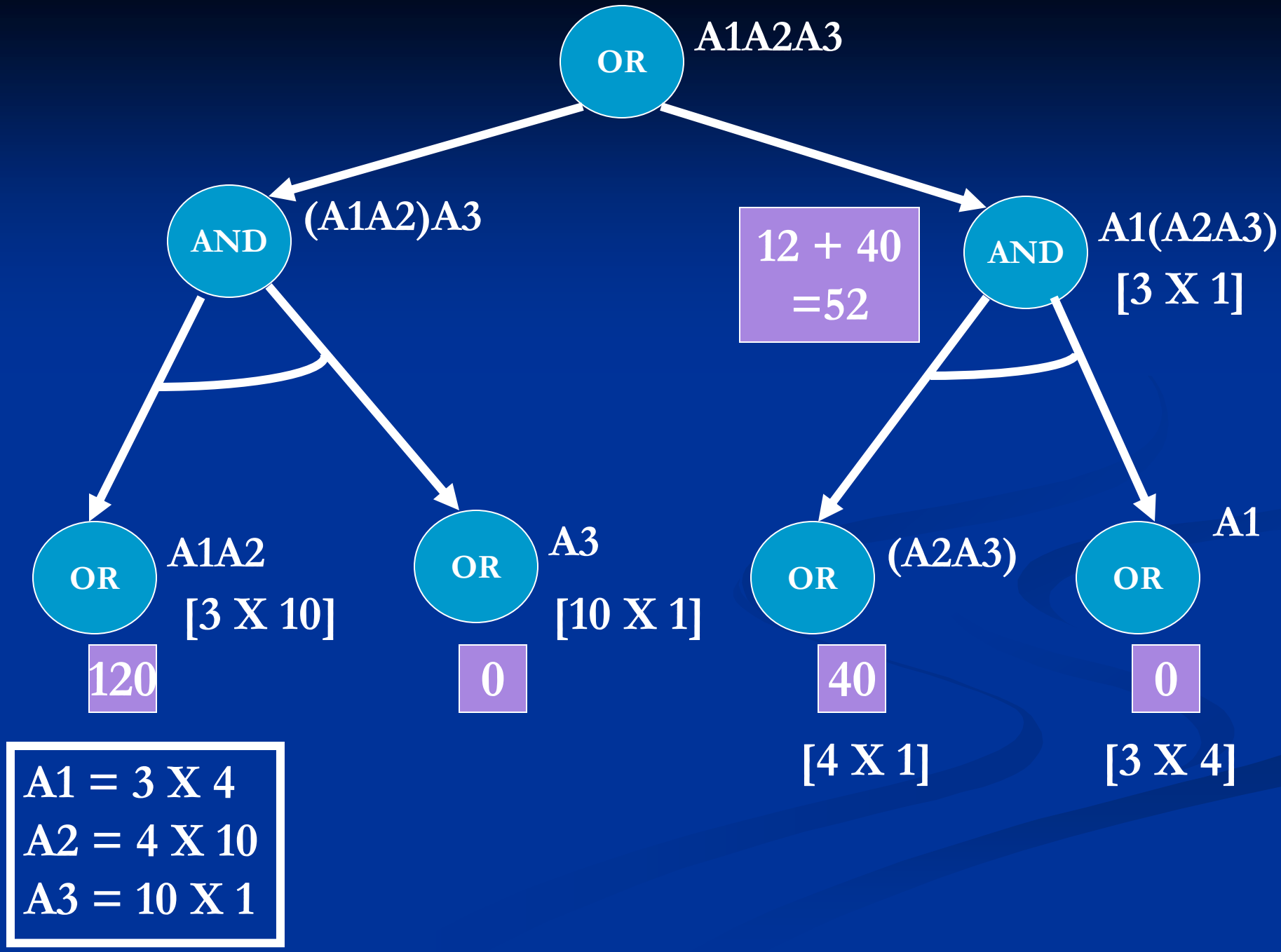
A1 = 3 X 4
A2 = 4 X 10
A3 = 10 X 1

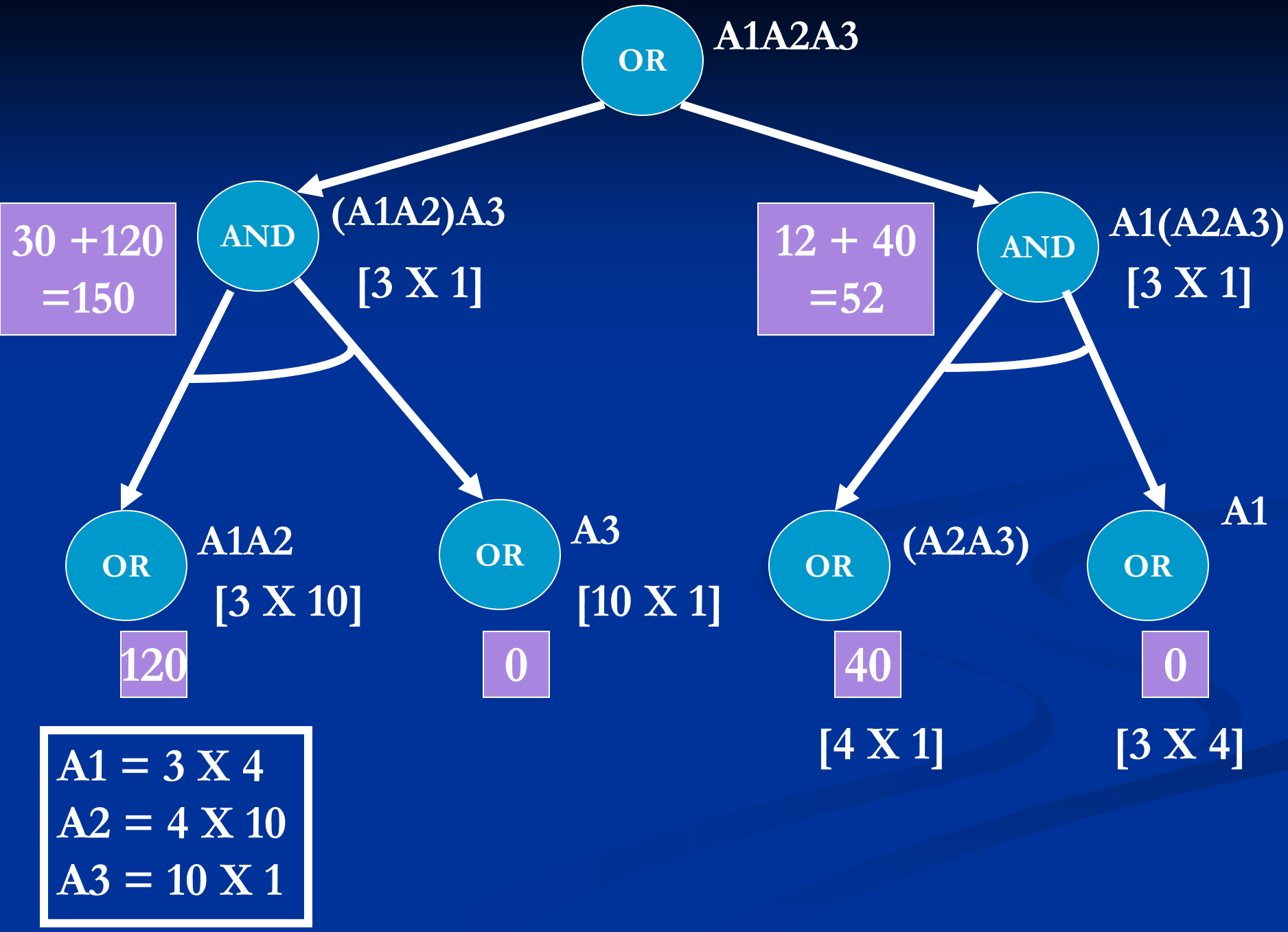


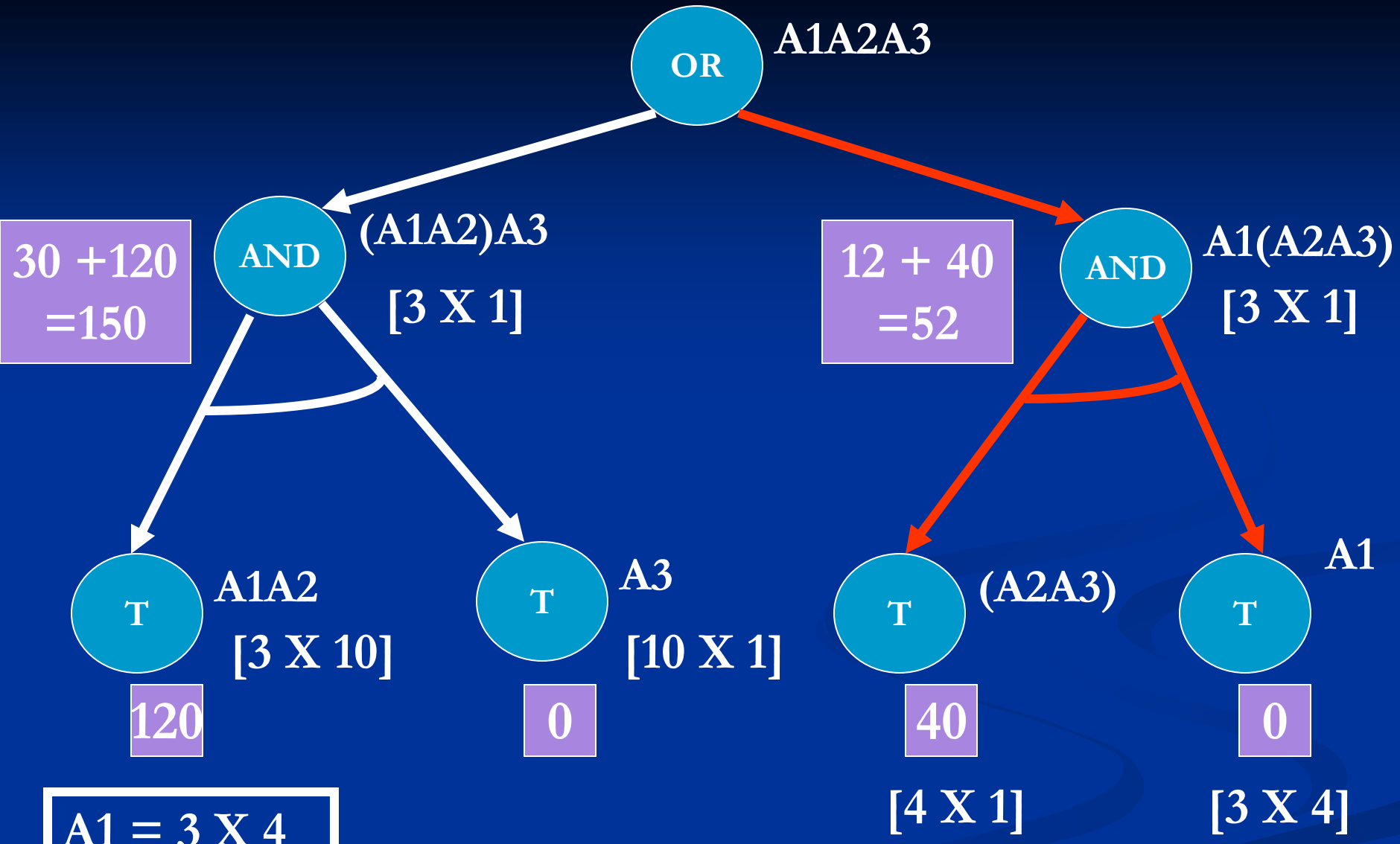
A1 = 3 X 4
A2 = 4 X 10
A3 = 10 X 1



A1 = 3 X 4
A2 = 4 X 10
A3 = 10 X 1


OR
A1A2A3
AND
(A1A2)A3
12 + 40
=52
AND
A1(A2A3)
[3 X 1]
OR
A1A2
[3 X 10]
120
OR
A3
[10 X 1]
0
OR
(A2A3)
40
[4 X 1]
OR
A1
0
[3 X 4]
A1 = 3 X 4
A2 = 4 X 10
A3 = 10 X 1


OR
A1A2A3
AND
(A1A2)A3
[3 X 1]
30 +120
=150
12 + 40
=52
AND
A1(A2A3)
[3 X 1]
OR
A1A2
[3 X 10]
OR
A3
[10 X 1]
OR
(A2A3)
OR
A1
120
0
40
0
[4 X 1]
[3 X 4]
A1 = 3 X 4
A2 = 4 X 10
A3 = 10 X 1



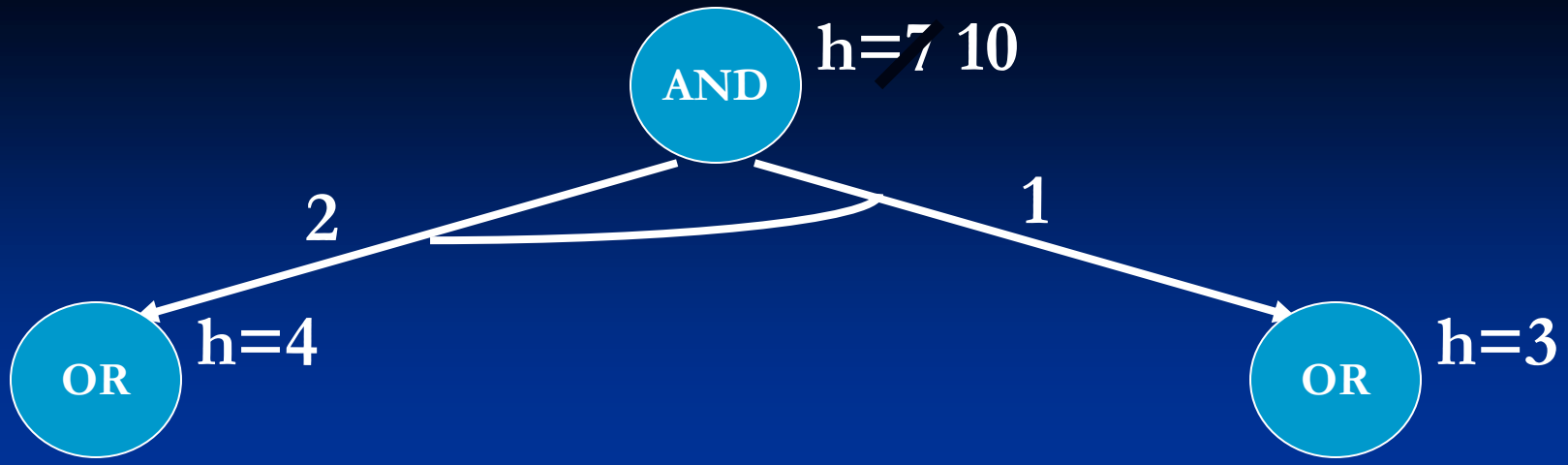
A1 = 3 X 4
A2 = 4 X 10
A3 = 10 X 1

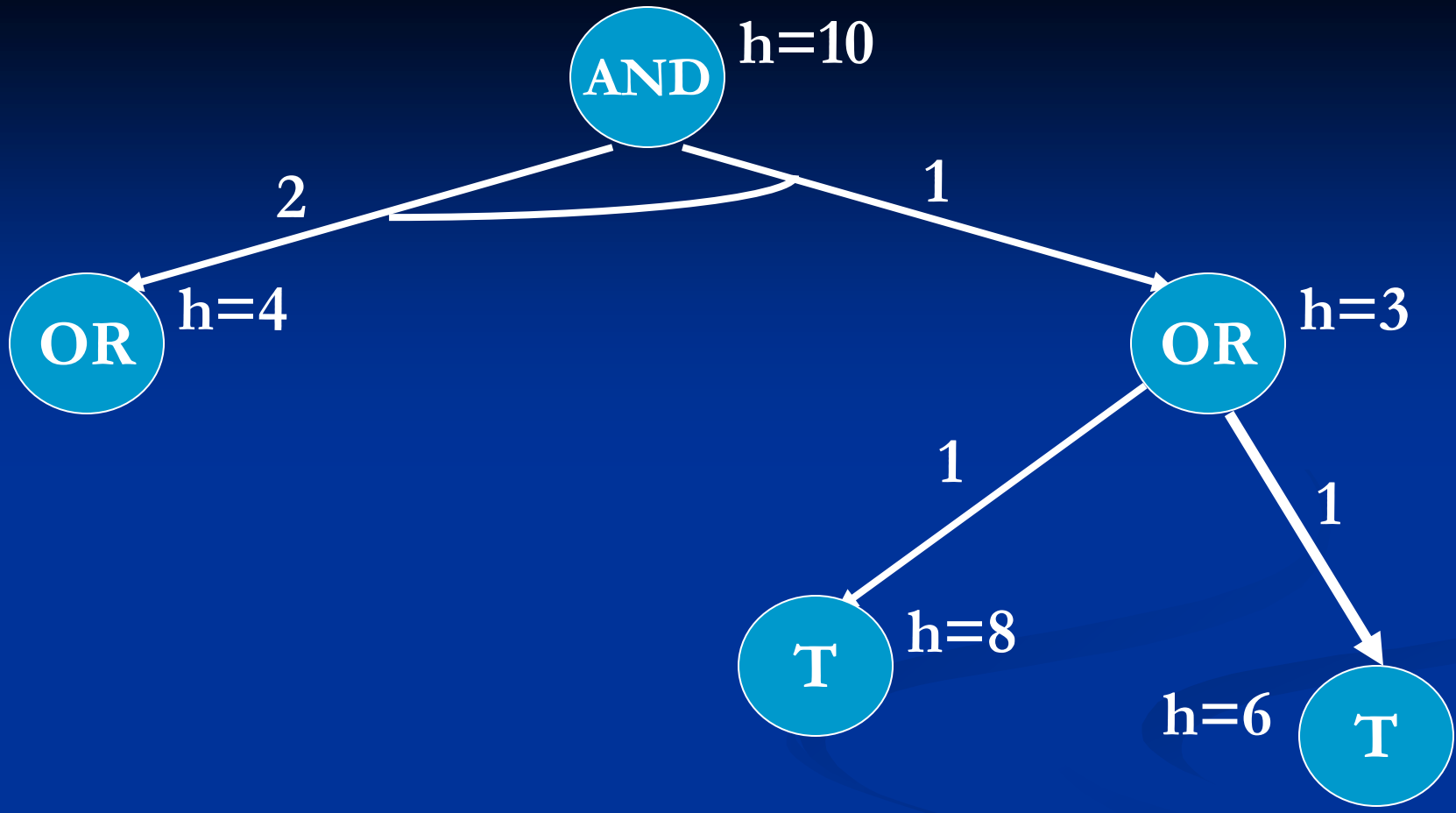
This is solved in the absence of heuristic function

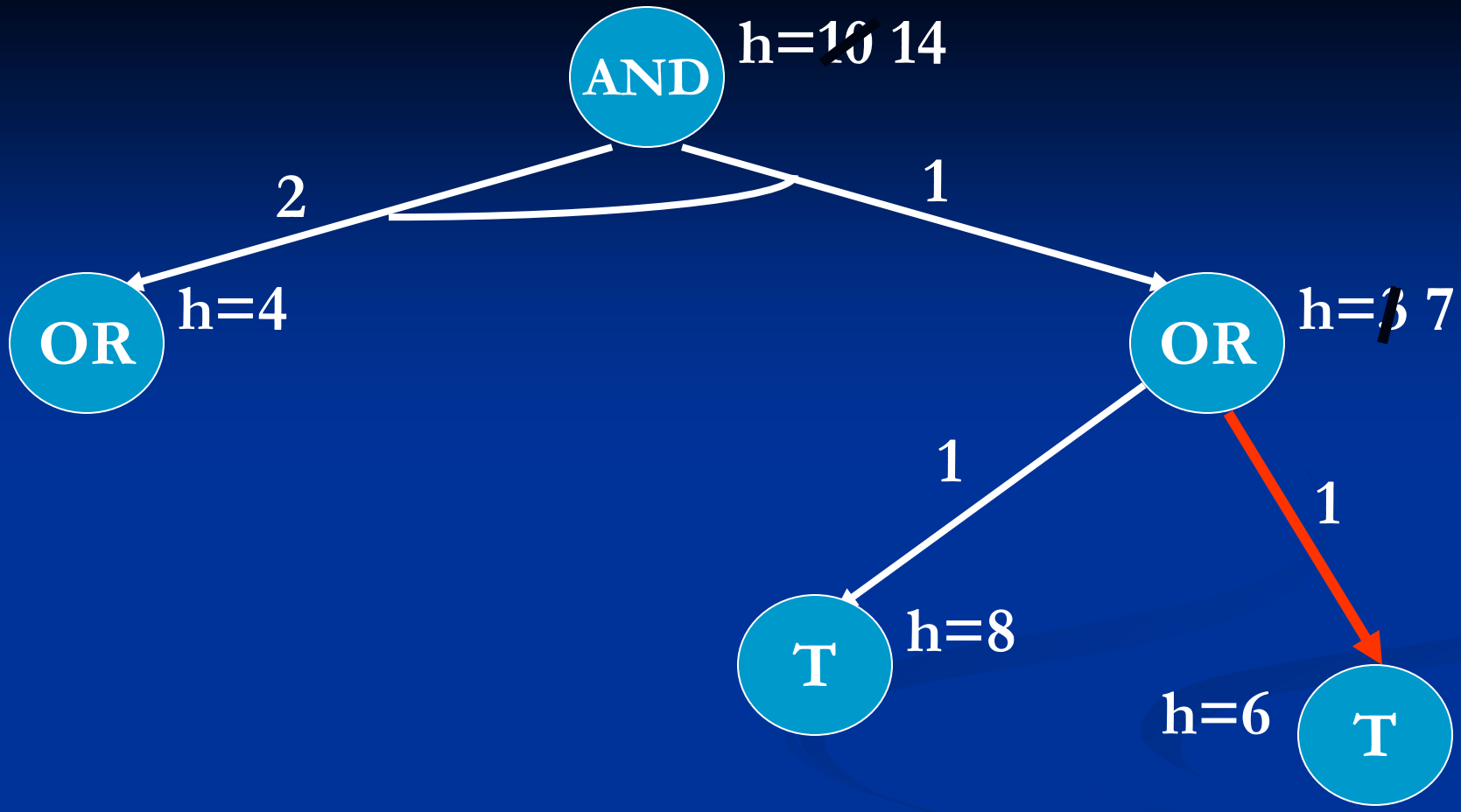
AND/OR Graph Search Problem

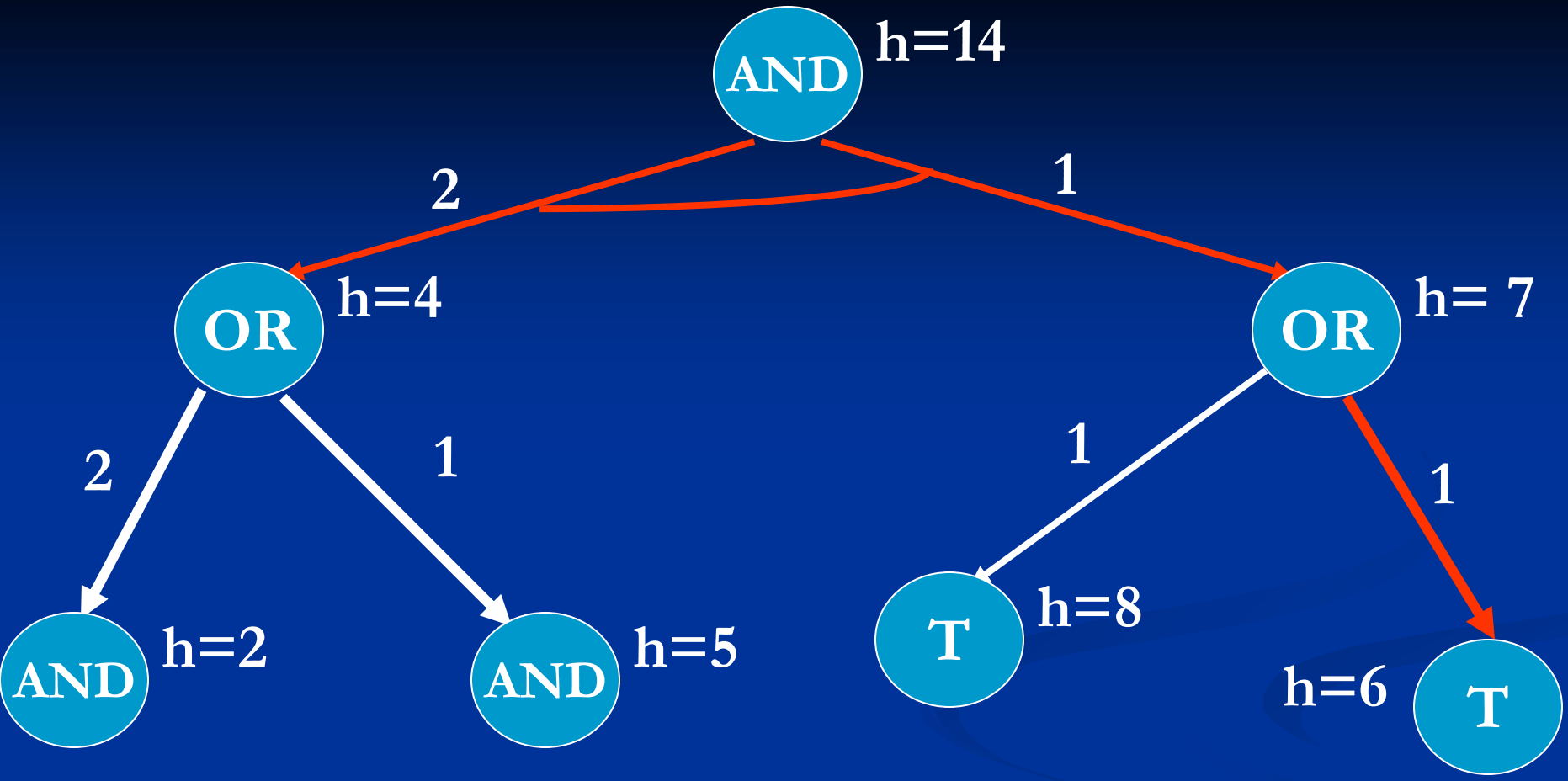
- **Problem definition:**
  - Given: [G, s, T, h] where:
    - G: implicitly specified AND/OR graph
    - s: start node of the AND/OR graph
    - T: set of terminal nodes
    - h(n): heuristic function estimating the cost of solving the sub problem at n
- To Find:
  - Minimum Cost Solution Tree

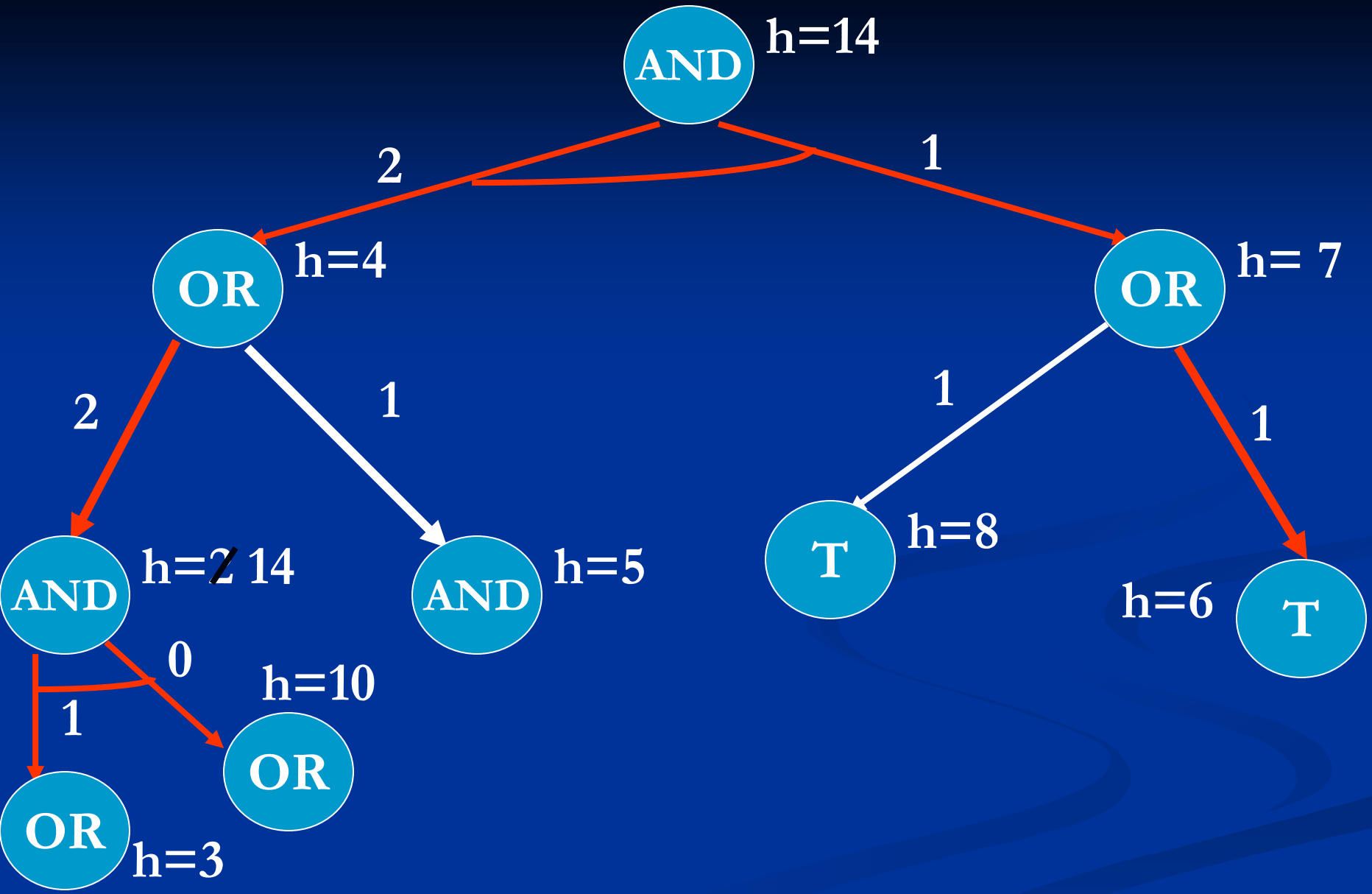
h=7

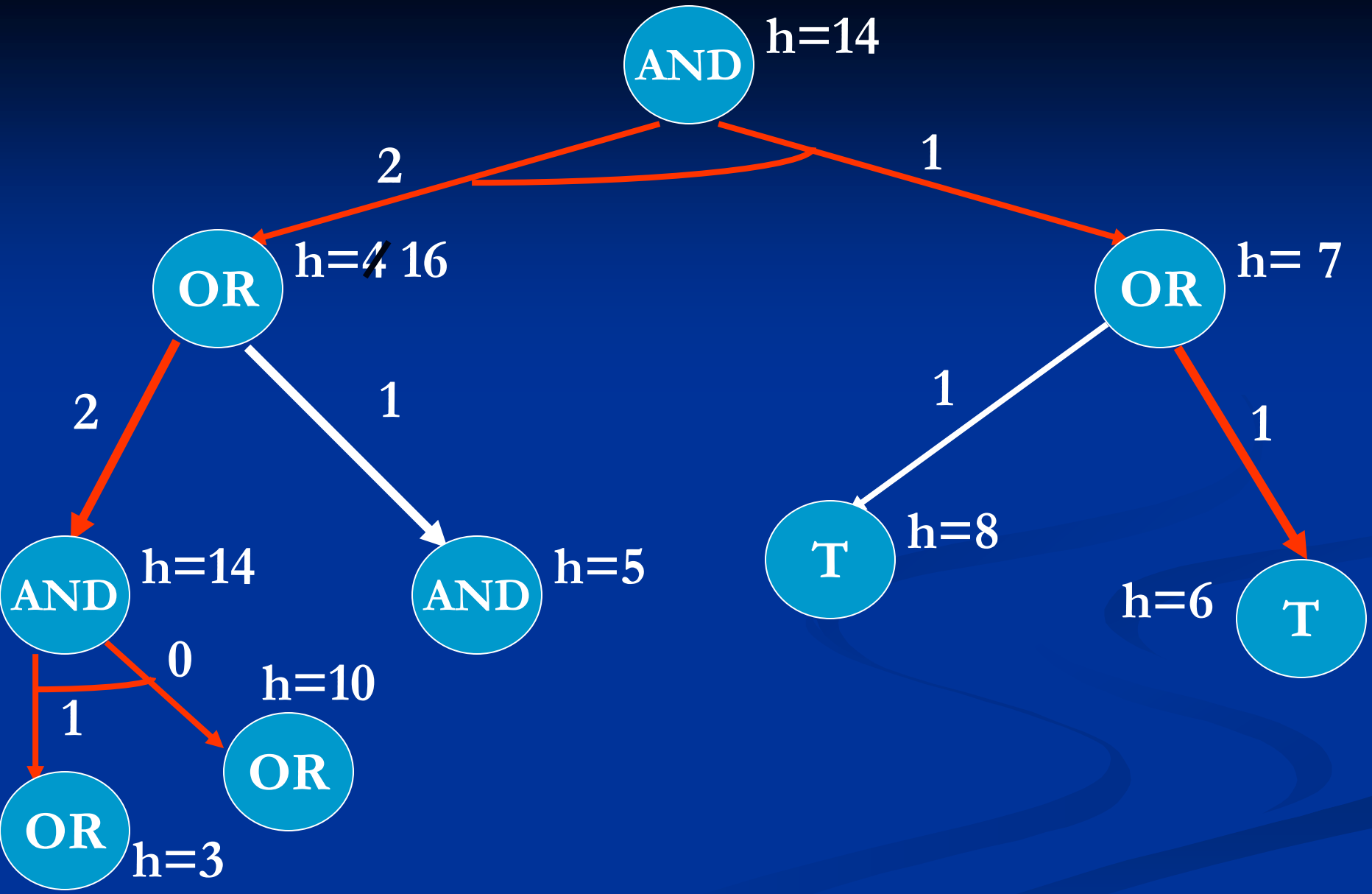

AND
h=7 10
2
1
OR
h=4
OR
h=3

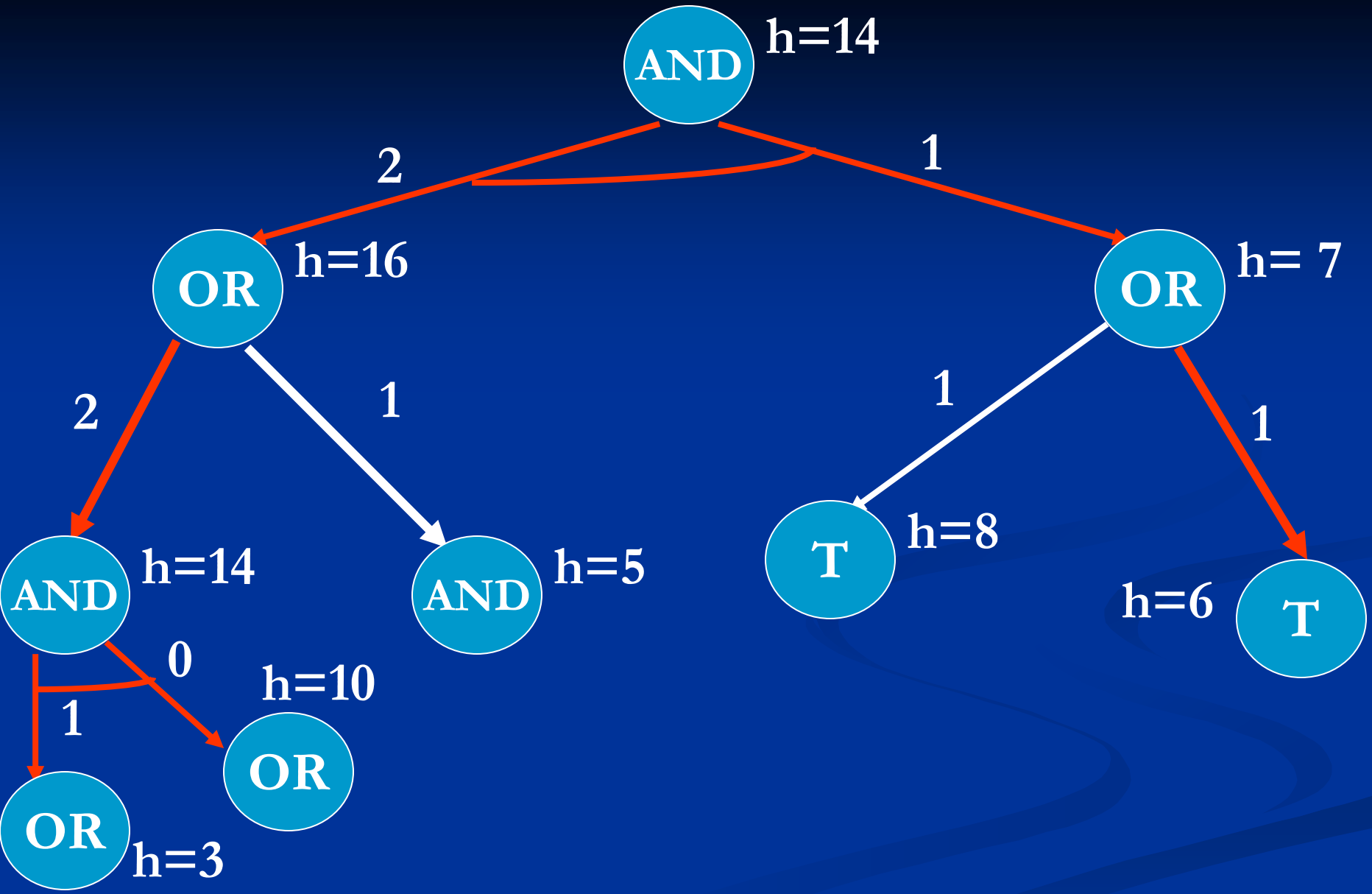

AND
h=10
2
1
OR
h=4
OR
h=3
1
1
T
h=8
h=6
T

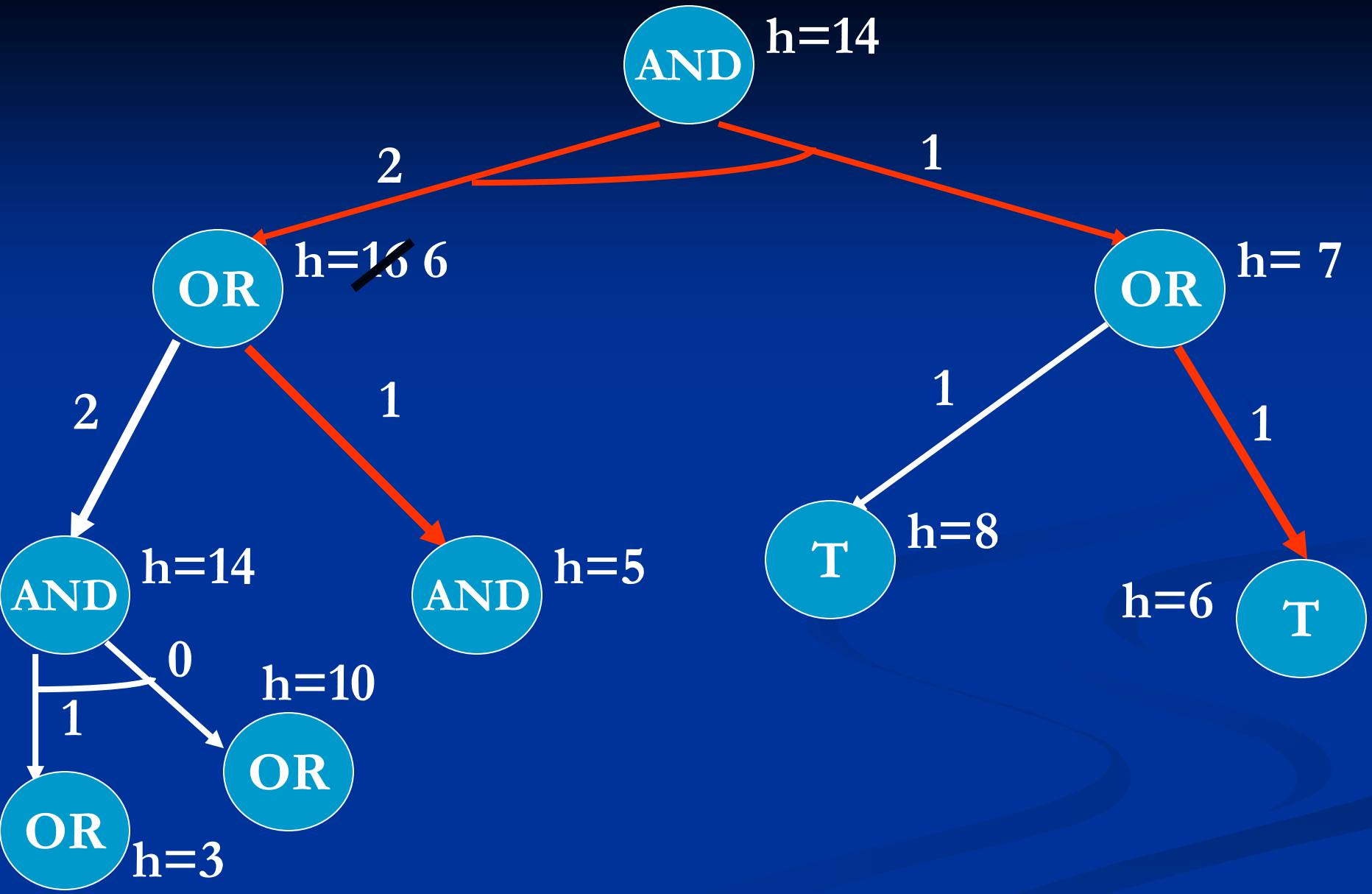

AND
h=14
2
1
OR
h=4
OR
h=7
1
1
T
h=8
h=6
T

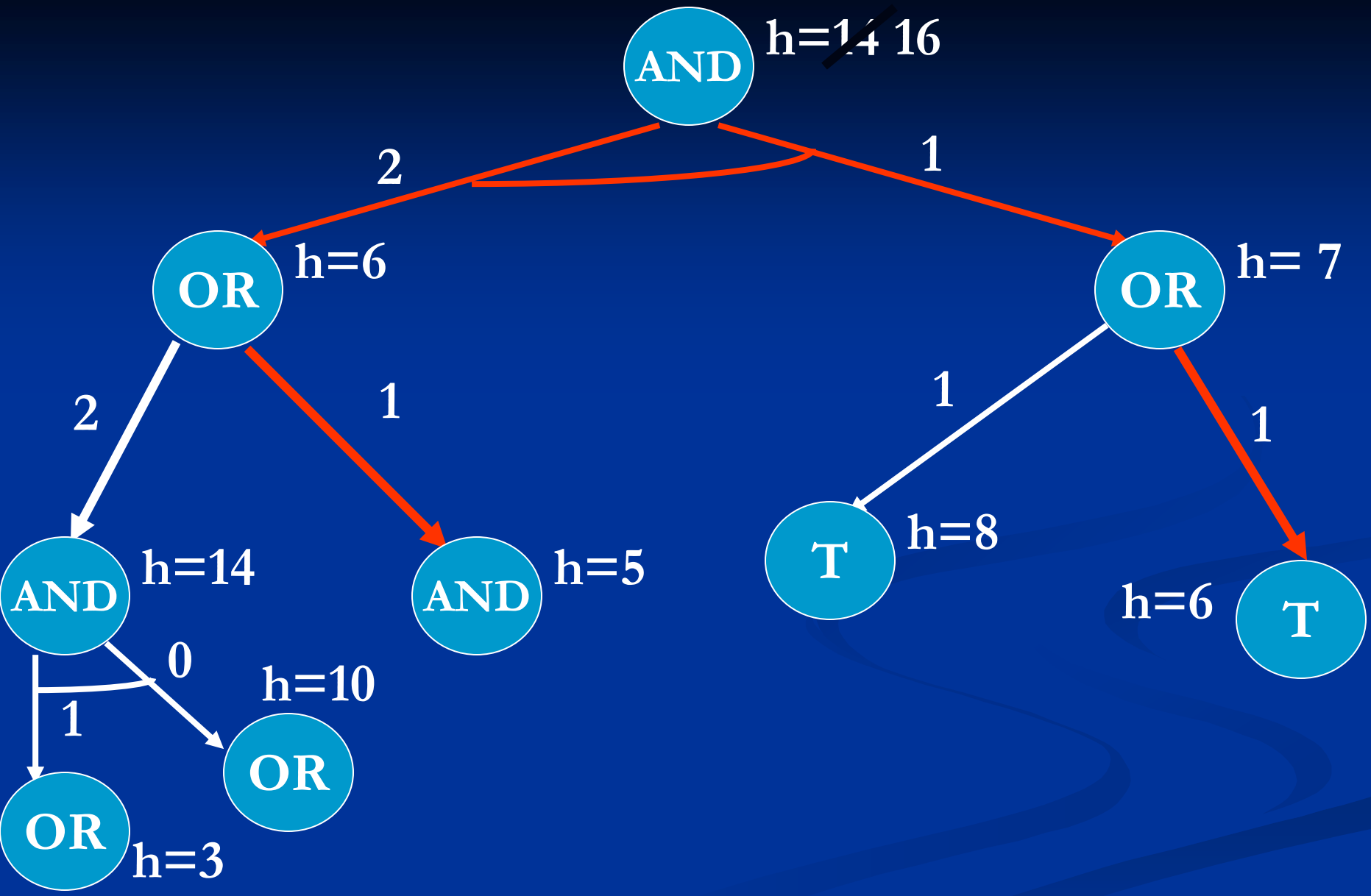

AND
h=14
2
1
OR
h=4
OR
h= 7
2
1
1
1
AND
h=2
AND
h=5
T
h=8
h=6
T

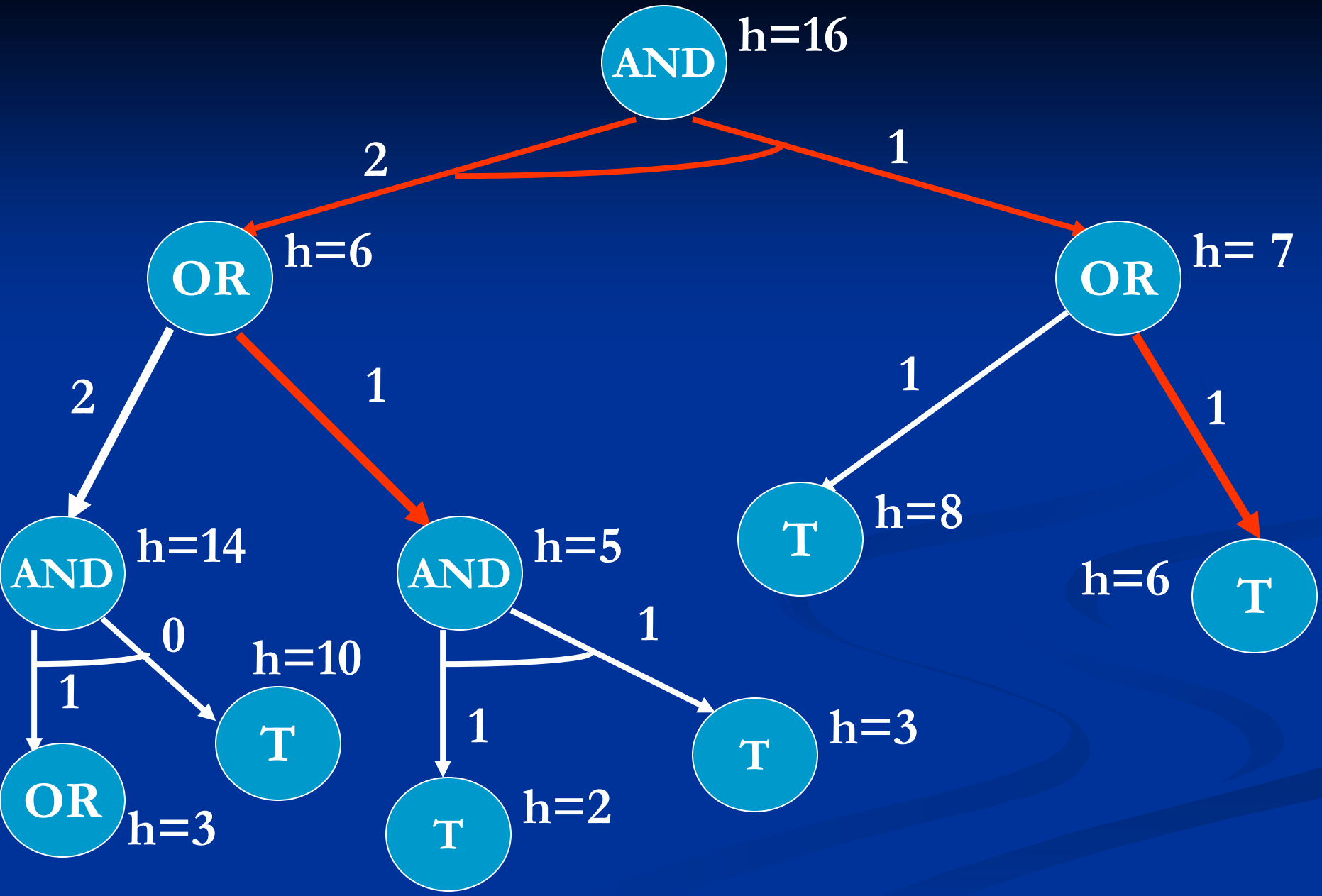

AND
h=14
2
1
OR
h=4
OR
h= 7
2
1
1
1
AND
h=2 14
AND
h=5
T
h=8
h=6
T
0
h=10
1
OR
OR
h=3

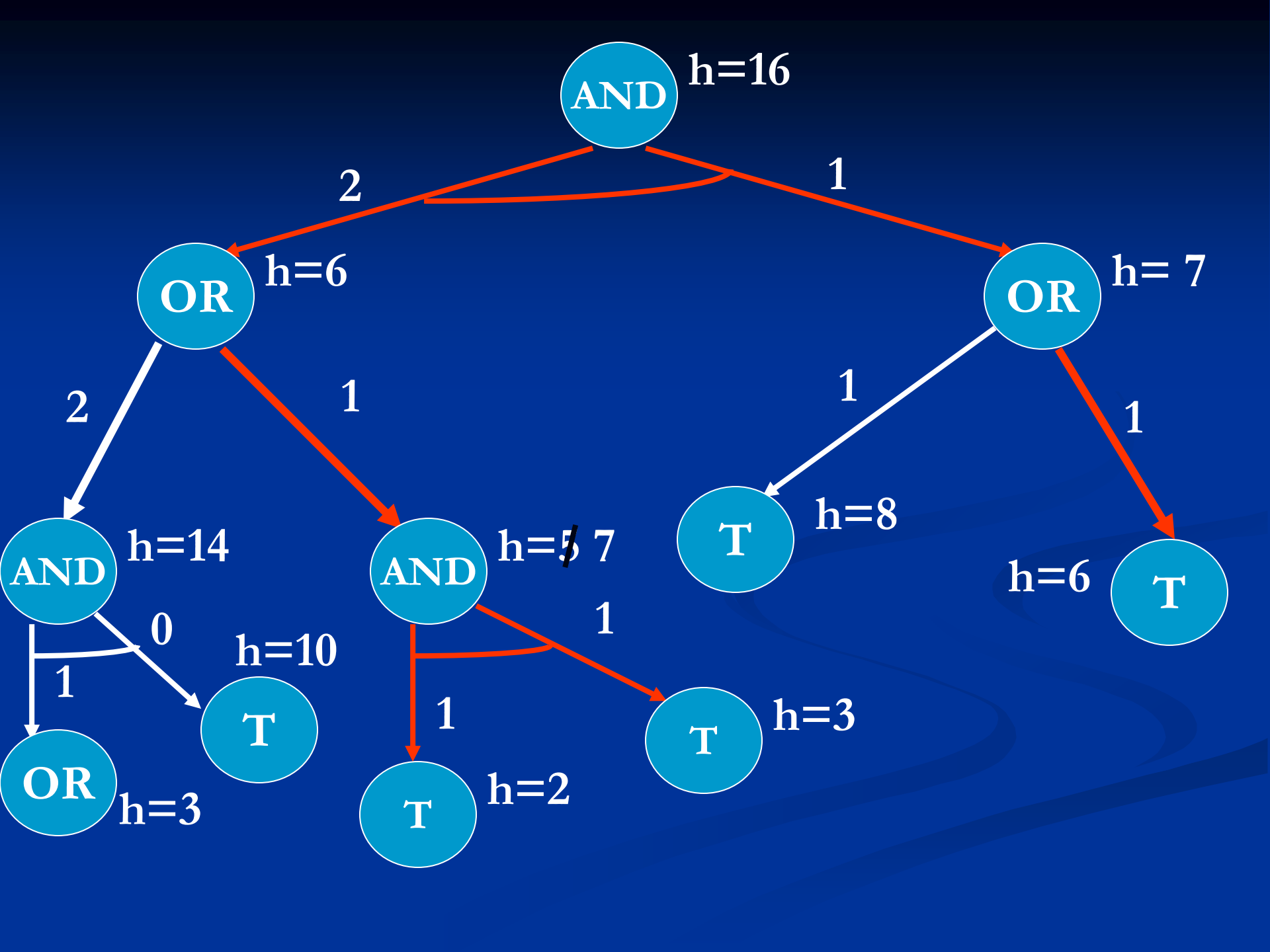

AND
h=14
2
1
OR
h=4 16
OR
h= 7
2
1
1
1
AND
h=14
AND
h=5
T
h=8
h=6
T
0
h=10
1
OR
OR
h=3

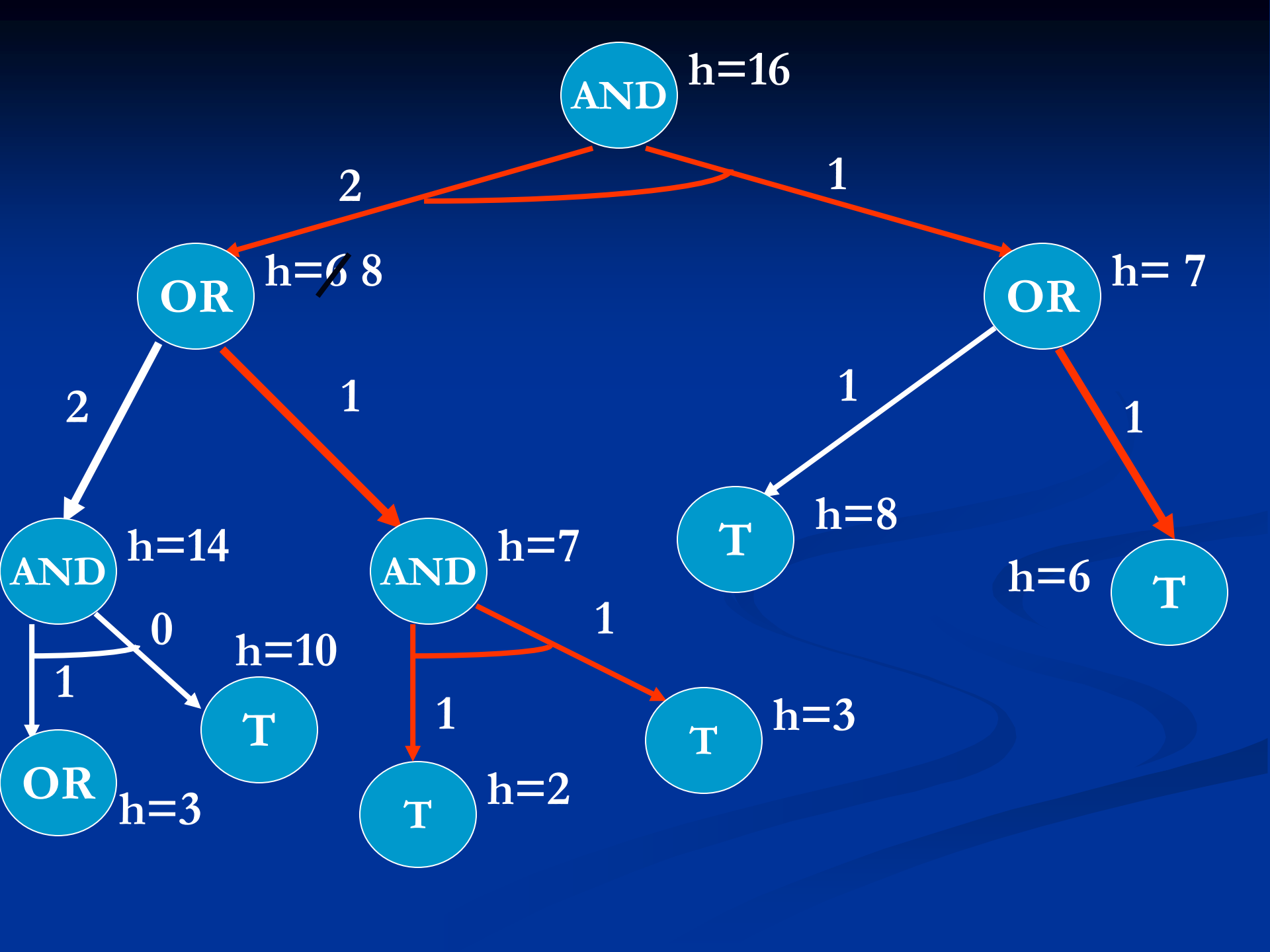

AND
h=14
2
1
OR
h=16
OR
h= 7
2
1
1
1
AND
h=14
AND
h=5
T
h=8
h=6
T
0
h=10
1
OR
OR
h=3

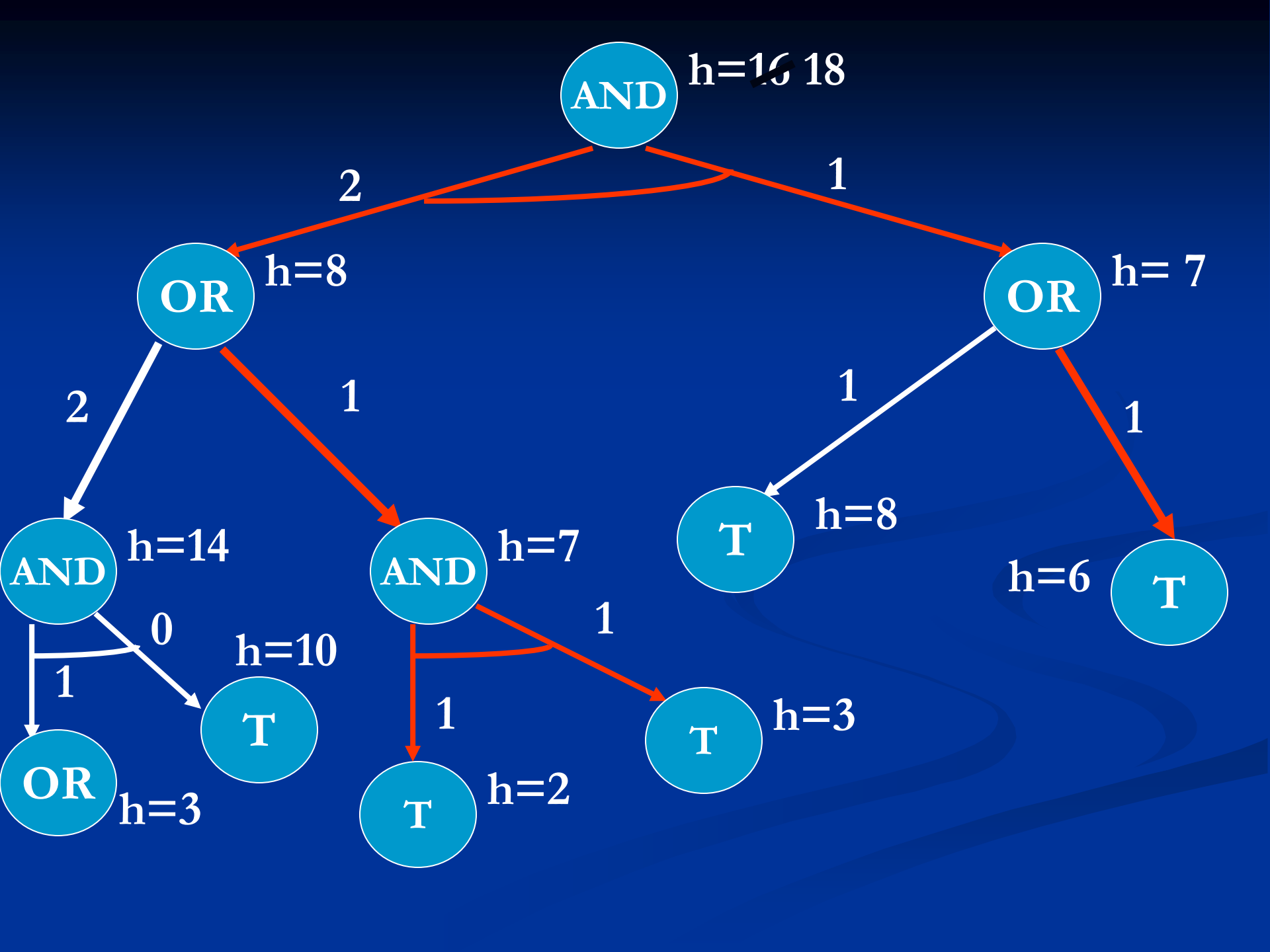

AND
h=14
2
1
OR
h=16 6
OR
h= 7
2
1
1
1
AND
h=14
AND
h=5
T
h=8
h=6
T
0
h=10
1
OR
OR
h=3

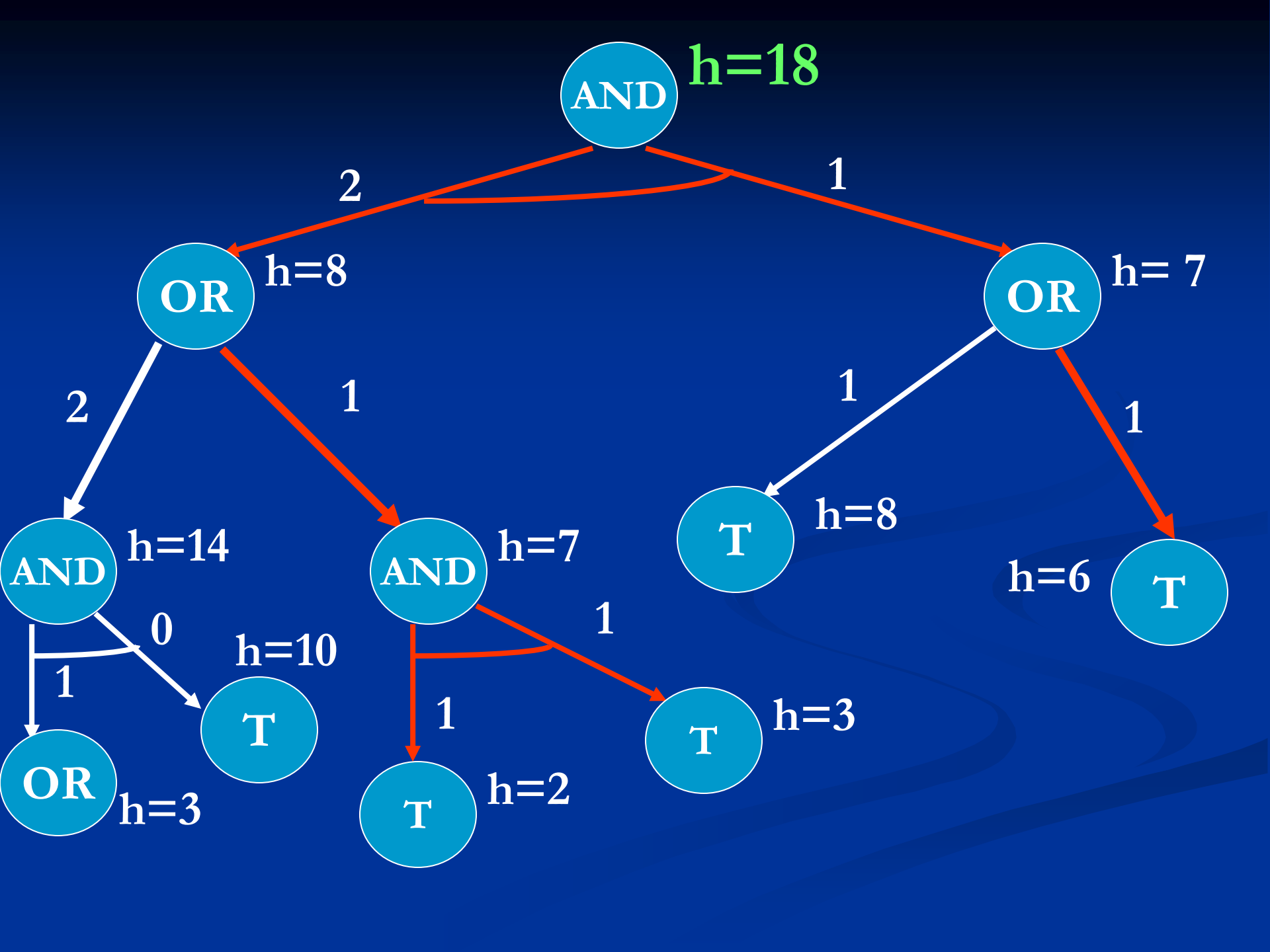

AND
h=14 16
2
1
OR
h=6
OR
h= 7
2
1
1
1
AND
h=14
AND
h=5
T
h=8
h=6
T
0
h=10
1
OR
OR
h=3

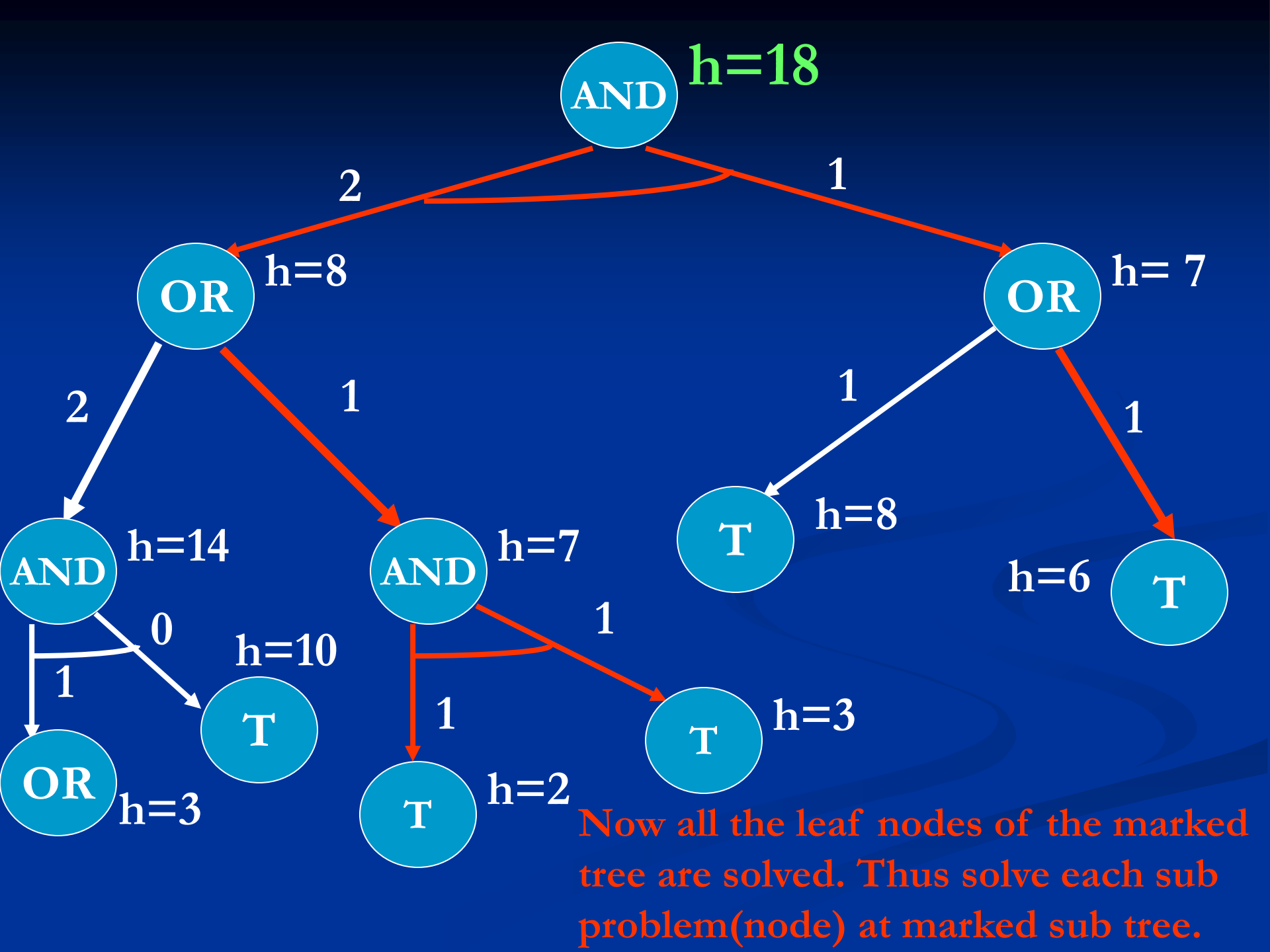

AND
h=16
2
1
OR
h=6
OR
h= 7
2
1
1
1
AND
h=14
AND
h=5
T
h=8
h=6
T
0
h=10
1
1
1
T
T
h=3
OR
h=3
T
h=2


AND
h=16
2
1
OR
h=6
OR
h= 7
2
1
1
1
AND
h=14
AND
h=5 7
T
h=8
h=6
T
0
h=10
1
1
T
T
h=3
OR
h=3
T
h=2


AND
h=16
2
1
OR
h=6 8
OR
h= 7
2
1
1
1
AND
h=14
AND
h=7
T
h=8
h=6
T
0
1
h=10
1
T
T
h=3
OR
h=3
T
h=2


AND
h=16 18
2
1
OR
h=8
OR
h= 7
2
1
1
1
AND
h=14
AND
h=7
T
h=8
h=6
T
0
1
h=10
1
T
T
h=3
OR
h=3
T
h=2


AND
h=18
2
1
OR
h=8
OR
h= 7
2
1
1
1
AND
h=14
AND
h=7
T
h=8
h=6
T
0
1
h=10
1
1
T
T
h=3
OR
h=3
T
h=2


AND
h=18
2
1
OR
h=8
OR
h= 7
2
1
1
1
AND
h=14
AND
h=7
T
h=8
h=6
T
0
1
h=10
1
1
T
T
h=3
OR
h=3
T
h=2
Now all the leaf nodes of the marked tree are solved. Thus solve each sub problem(node) at marked sub tree.

AO* Algorithm

1. **Initialize:** Set $G^* = \{s\}$, $f(s) = h(s)$
   if $s \in T$, label s as SOLVED
2. **Terminate:** if s is SOLVED, then Terminate
3. **Select:** A non-terminal leaf node n from the marked sub tree.
4. **Expand:** Mark explicit the successors of n
   For each new successor m:
   set $f(m) = h(m)$
   if $m \in T$, label m as SOLVED

AO* Algorithm

5. **Cost Revision:** Call cost-revise(n)
6. **Loop:** Goto step 2

Cost Revisoin in AO*
cost-revise(n)

1. Create Z={n}
2. If Z = { } then return
3. Select the node m from Z such that m has no descendants in Z
4. If m is an AND node with successors R1, R2, R3, …….. Rk

   set f(m) = Σ [f(Ri) + C(m,Ri)]

   Mark the edge to each successor of m

   if each successor is labeled SOLVED then label m as SOLVED

Cost-revise(n)

5. If m is an OR node with successors R1, R2, R3, …….. Rk

   set f(m) = min [f(Ri) + C(m,Ri)]

   Mark the edge to best successor of m

   if marked successor is labeled SOLVED then label m as SOLVED

6. If the cost or label of m has changed, then insert those parents of m into Z for which m is a marked successor

7. Goto step 2

Searching OR Graph

- How would AO* behave when we have only OR nodes ????