

AIFA

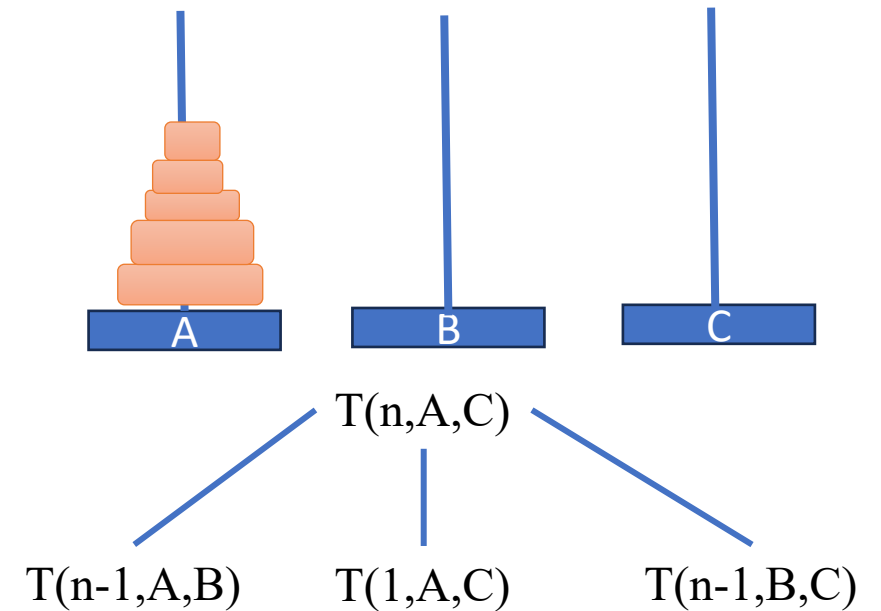
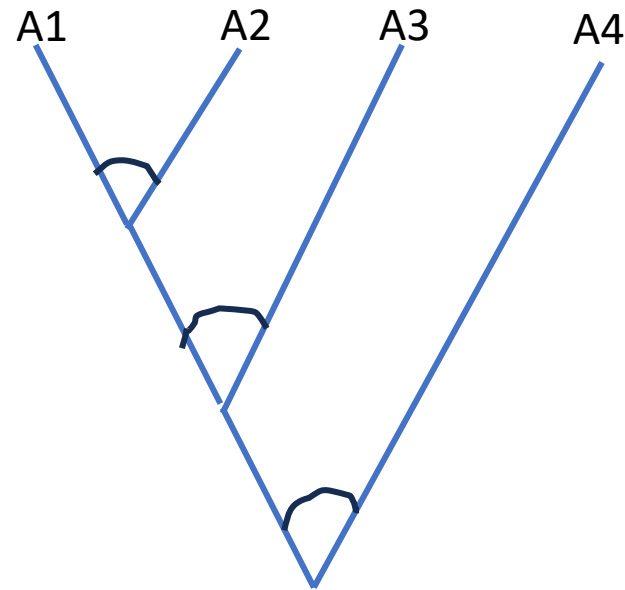
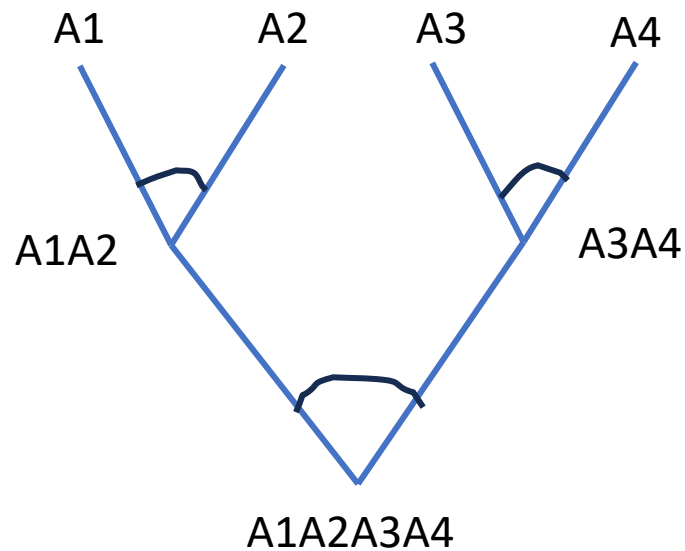
Problem Reduction Search

18/01/2024

Koustav Rudra

Problem Reduction Search

- Planning how best to solve a problem that can be recursively decomposed into sub-problems in multiple ways
 - Matrix multiplication problem
 - Tower of Hanoi
 - Theorem proving



Formulation

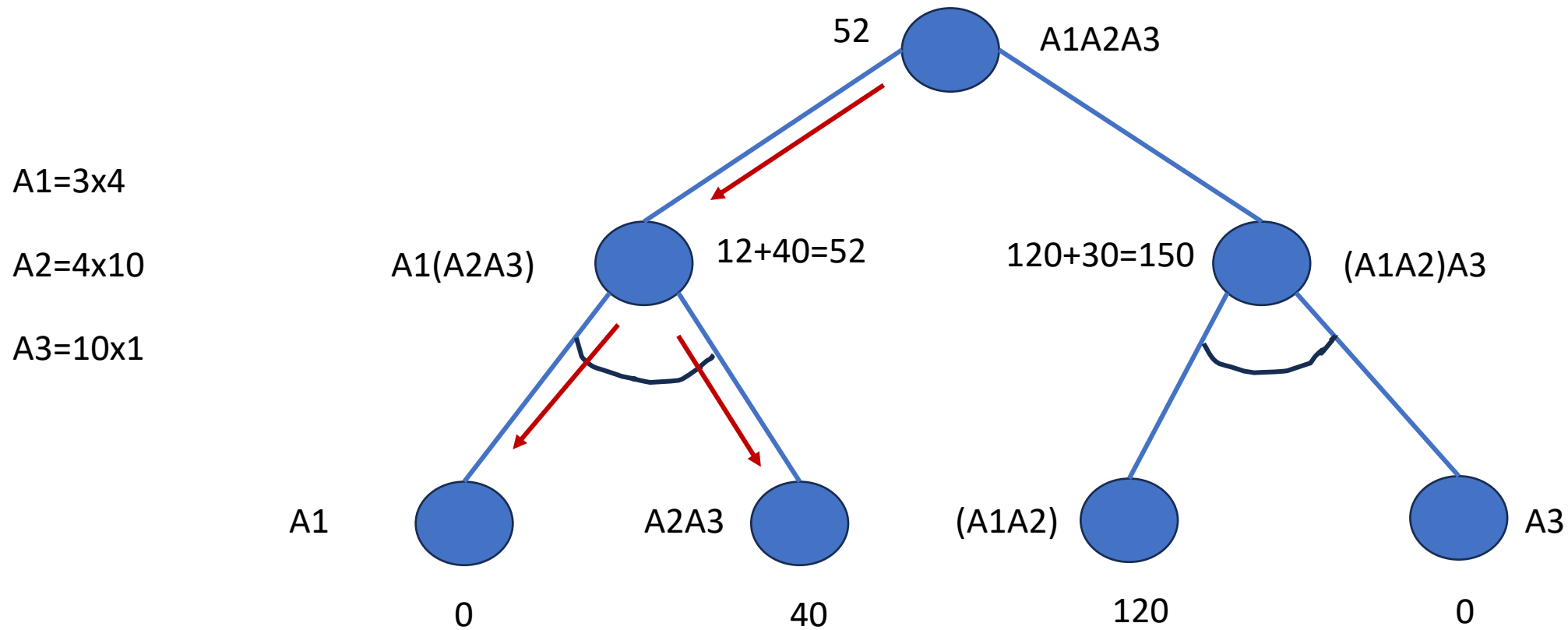
- AND/OR Graph

- An OR node represents a choice between possible decompositions
- An AND node represents a given decomposition

- Game Trees

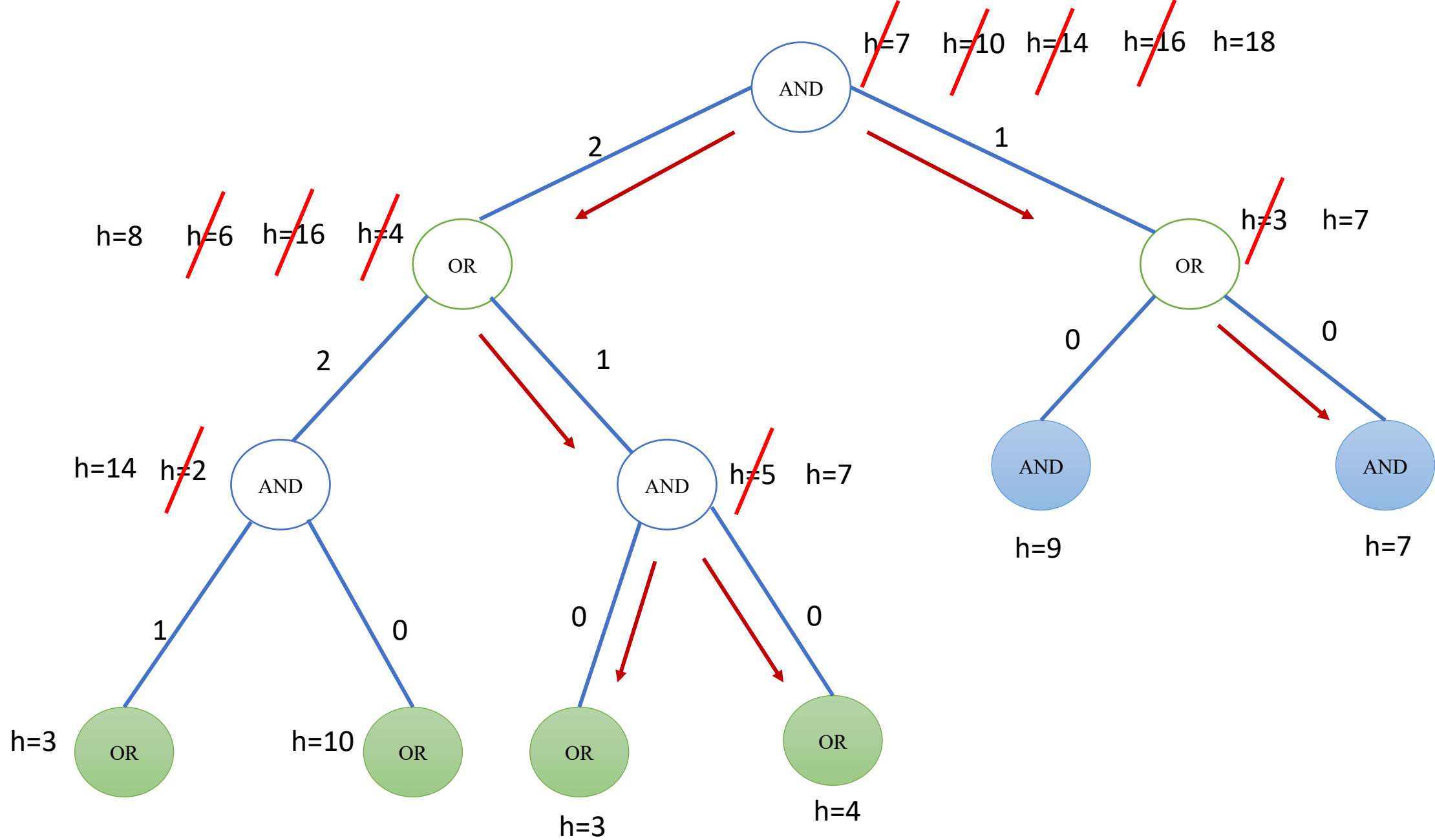
- Max/Min nodes
- Max nodes represent the choice of my opponent
- Min nodes represent my choice

Each node has a separate optimization criteria



- This is when heuristics is not present

AO*: Example



Algorithm AO*

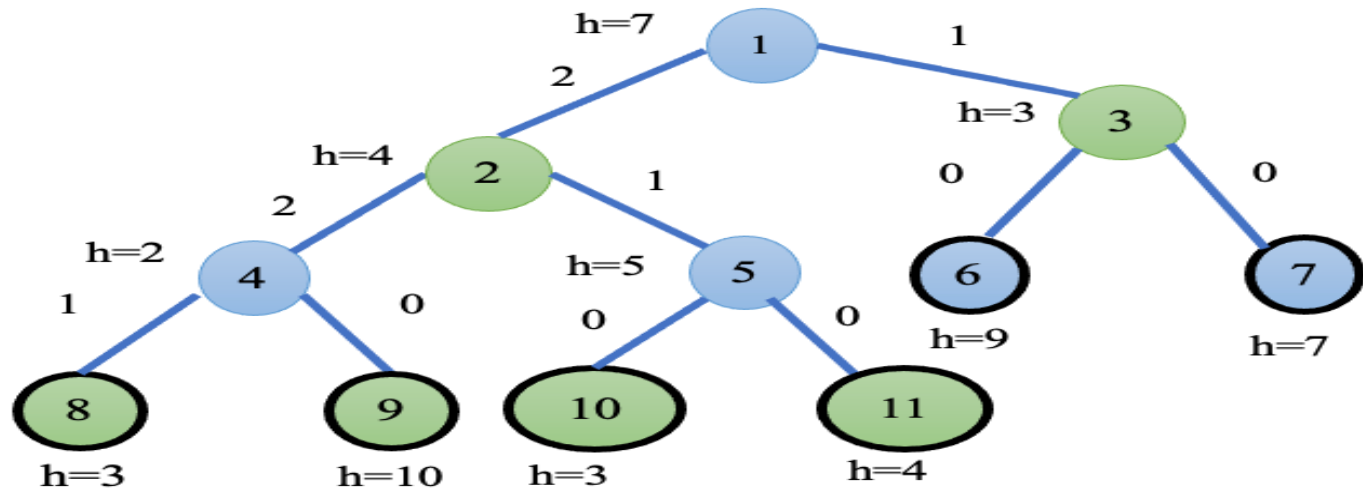
- **Initialize:** Set $G^* = \{s\}$, $f(s) = h(s)$
 - If $s \in T$, terminate and label s as solved
- **Terminate:**
 - If s is solved, then Terminate
- **Select:** Select a nonterminal leaf node n from the marked subtree
- **Expand:**
 - Make explicit the successors of n
 - For each new successor, m :
 - Set $f(m) = h(m)$
 - If m is terminal, label m solved
- **Cost revision:** Call cost-revision(n)
- **Loop:** Go to Step 2

Cost Revision in AO*: cost-revise(n)

- Create $Z = \{n\}$
- If $Z = \{\}$ return
- Select a node m from Z such that m has no descendants in Z
- If m is an AND node with successors r_1, r_2, \dots, r_k :
 - Set $f(m) = \sum(f(r_i) + C(m, r_i))$
 - Mark the edge to each successor of m
 - If each successor is labeled SOLVED
 - Then label m as SOLVED

Cost Revision in AO*: cost-revise(n)

- If **m** is an **OR node** with successors r_1, r_2, \dots, r_k :
 - Set $f(m) = \min\{f(r_i) + C(m, r_i)\}$
 - Mark the edge to the best successor of m
 - If the **marked successor** is **labeled SOLVED**
 - Then label **m** as **SOLVED**
- If the cost of label m has changed,
 - Then insert those parents of m into Z for which m is a marked successor

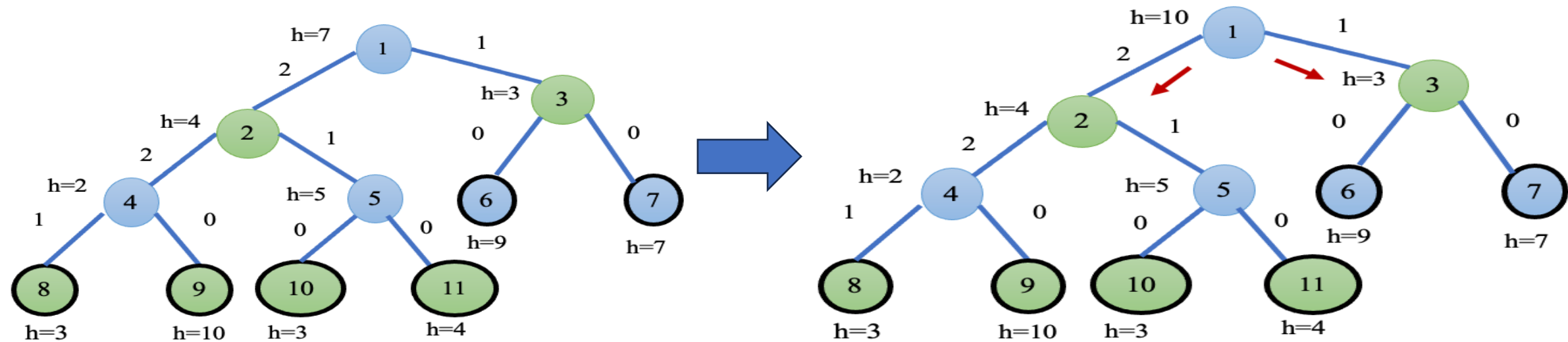


OPEN SET		SELECT	TER	EXPANDED	COST REVISION
----------	--	--------	-----	----------	---------------

[1(7)]		1(7)	N	[2(4,N),3(3,N)]	[1(7)]
--------	--	------	---	-----------------	--------

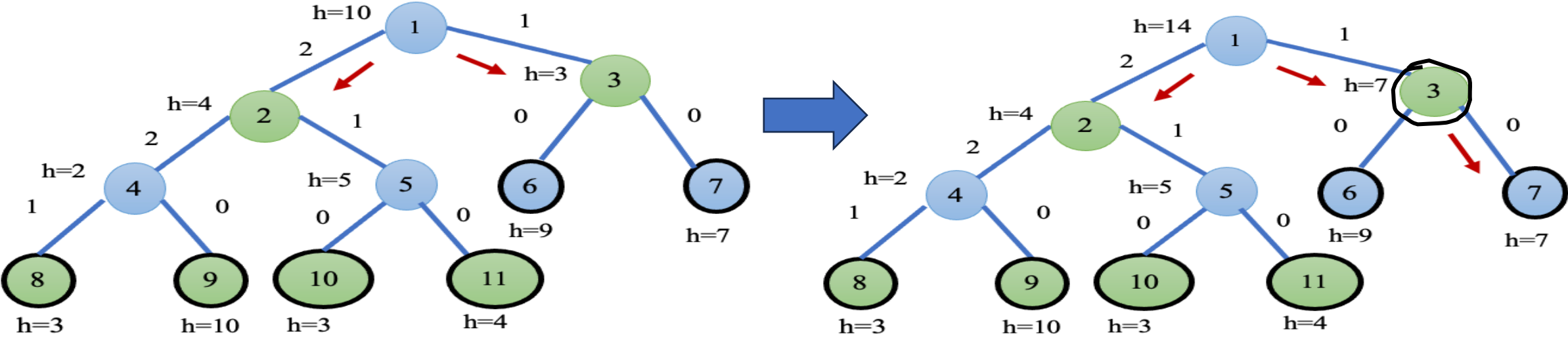
START LIST	RET	SELECT	UPDATE	EDGE MARK	SOLVED	NEW LIST
------------	-----	--------	--------	-----------	--------	----------

[1(7)]	N	1(7)	[1(10)]	[(1,2),(1,3)]	N	[]
--------	---	------	---------	---------------	---	----



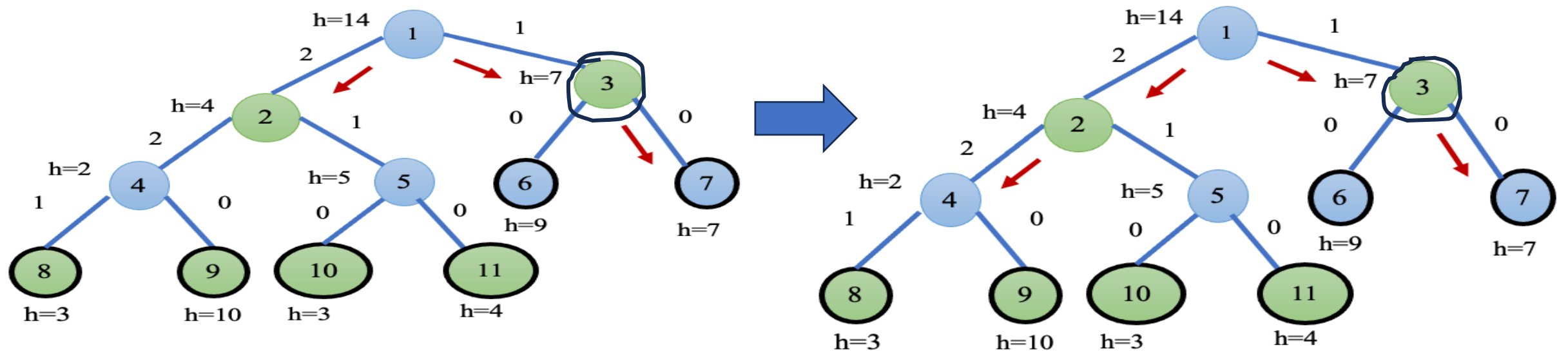
OPEN SET		SELECT	TER	EXPANDED	COST REVISION
[2(4),3(3)]		3(3)	N	[2(4,N),6(9,S),7(7,S)]	[3(3)]

START LIST	RET	SELECT	UPDATE	EDGE MARK	SOLVED	NEW LIST
[3(3)]	N	3(3)	[3(7)]	[(3,7)]	S	[1(10)]
[1(10)]	N	1(10)	[1(14)]	[(1,2),(1,3)]	N	[]



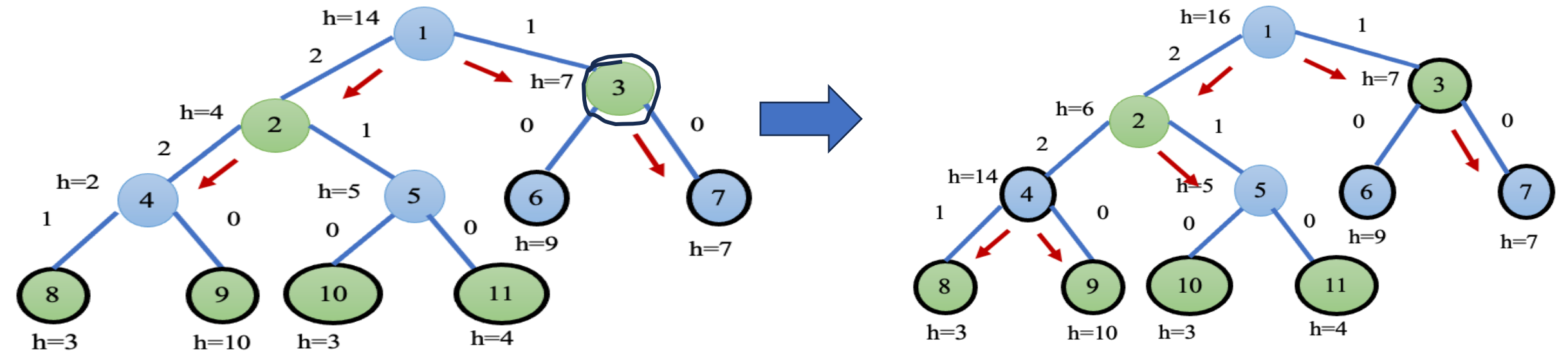
OPEN SET		SELECT	TER	EXPANDED	COST REVISION	
[2(4),6(9),7(7)]		2(4)	N	[4(2,N),5(5,N),6(9,S),7(7,S)]	[2(4)]	

START LIST	RET	SELECT	UPDATE	EDGE MARK	SOLVED	NEW LIST
[2(4)]	N	2(4)	[2(4)]	[(2,4)]	N	[]



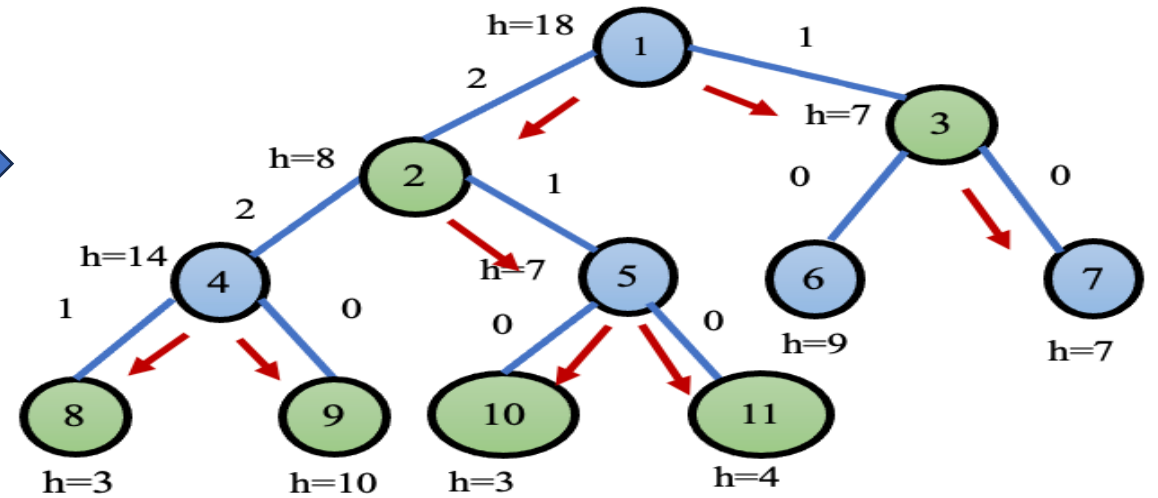
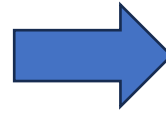
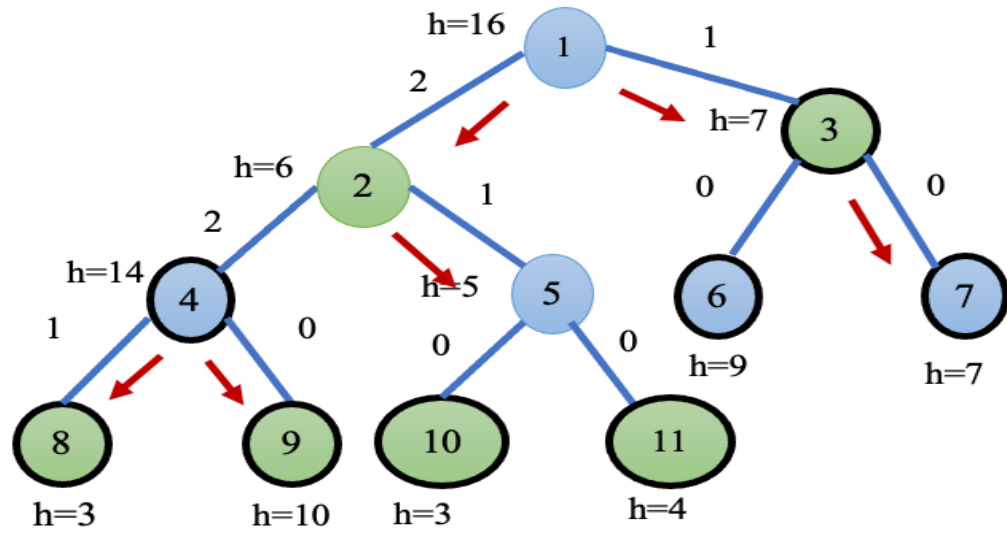
OPEN SET		SELECT	TER	EXPANDED	COST REVISION
[4(2),5(5),6(9),7(7)]		4(2)	N	[8(3,S),9(10,S),5(5,N),6(9,S),7(7,S)]	[4(2)]

START LIST	RET	SELECT	UPDATE	EDGE MARK	SOLVED	NEW LIST
[4(2)]	N	4(2)	[4(14)]	[(4,8),(4,9)]	S	[2(4)]
[2(4)]	N	2(4)	[2(6)]	[(2,5)]	N	[1(14)]
[1(14)]	N	1(14)	[1(16)]	[(1,2),(1,3)]	N	[]



OPEN SET		SELECT	TER	EXPANDED	COST REVISION
[8(3),5(5),9(10),6(9),7(7)]		5(5)	N	[8(3,S),9(10,S),10(3,S),11(4,S),6(9,S),7(7,S)]	[5(5)]

START LIST	RET	SELECT	UPDATE	EDGE MARK	SOLVED	NEW LIST
[5(5)]	N	5(5)	[5(7)]	[(5,10),(5,11)]	S	[2(6)]
[2(6)]	N	2(6)	[2(8)]	[(2,5)]	S	[1(16)]
[1(16)]	N	1(16)	[1(18)]	[(1,2),(1,3)]	S	[]



OPEN SET		SELECT	TER	EXPANDED	COST REVISION	
[8(3),9(10),10(3),11(4),6(9),7(7)]		N	Y			
START LIST	RET	SELECT	UPDATE	EDGE MARK	SOLVED	NEW LIST

Thank You