Student Name: ROHITH PEDDI Student NetID:RXP190007

University of Texas at Dallas
Department of Computer Science
CS 6364.001 – Artificial Intelligence
Fall 2020
Instructor: Dr. Sanda Harabagiu

Instructions: Do not communicate with anyone in any shape or form. This is an independent exam. Do not delete any problem formulation, just attach your answer in the space provided. If the problem is deleted and you send only the answer, you shall receive ZERO points.
Copy and paste the Mid-Term Exam into a Word document, enter your answers (either by typing in Word, or by inserting a VERY CLEAR picture of your hand-written solution) and transform the file of the exam into a PDF format. If we cannot clearly read the picture, you will get ZERO for that answer! Make sure that you insert EACH answer immediately after EACH question. Failure to do so will result in ZERO points for the entire exam! Submit the PDF file with the name **MidTerM_Exam_netID.pdf**, where netID is your unique netid provided by UTD. If you submit your exam in any other format your will receive ZERO points. The Midterm shall be submitted in eLearning before the deadline. No late submissions shall be graded! Any cheating attempt will determine the ENTIRE grade of the mid-term to become ZERO.

Problem 1 (50 points)

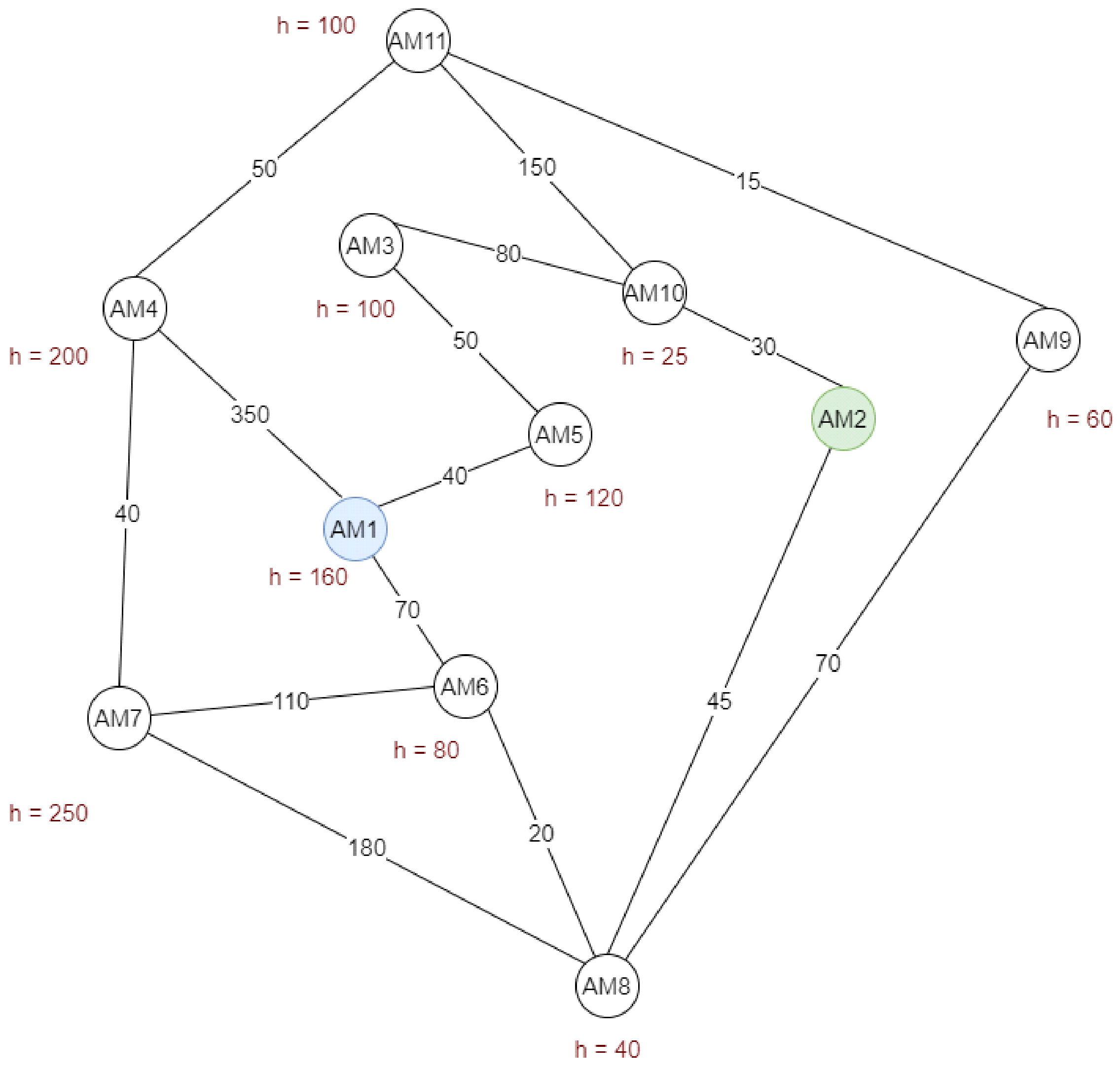
Proteins have an amino acid "alphabet" of 11 elements: AM1, AM2, ..., AM11. *Amino acids are chemically linked together to form protein chains. Between amino acids there are chemical links of different strengths. Suppose you examine under microscope a sample of a protein that belongs to an alien species, having only 11 amino acids. You want to generate an optimal path between* AM1 *and* AM2 *using the A* search algorithm. You are given the strengths of the chemical links in the sample as a graph representation:*

Oracle distance to AM2		The Graph			
AM1	160	AM11	-----	AM4	:::: 50
AM3	100	AM11	-----	AM10	:::: 150
AM4	200	AM11	-----	AM9	:::: 15
AM5	120	AM4	-----	AM7	:::: 40
AM6	80	AM7	-----	AM8	:::: 180
AM7	250	AM7	-----	AM6	:::: 110
AM8	40	AM9	-----	AM8	:::: 70
AM9	60	AM10	-----	AM2	:::: 30
AM10	25	AM8	-----	AM2	:::: 45
AM11	100	AM10	-----	AM3	:::: 80
******************************		AM3	-----	AM5	:::: 50
******************************		AM5	-----	AM1	:::: 40
******************************		AM1	-----	AM6	:::: 70
******************************		AM6	-----	AM8	:::: 20
******************************		AM1	-----	AM4	:::: 350
******************************					

An oracle also gives you the heuristic distance values to AM2 *from each other amino acid in the sample. This heuristic is consistent. Specify if you will use TREE-SEARCH or GRAPH-SEARCH. Motivate your decision.* **(5 points)**
Provide the path of amino acids from AM1 *to* AM2 *as well as the cost of obtaining it. it. Describe at each step of the search (1) what amino acids you have on the search frontier; (2) the current list of explored amino acids; (3) the current path from AM1 to the current amino acid and the cost of that path. Show the successors of each current node, show how you compute all the evaluation functions and which node you select for the next step.* **(45 points)**

SOLUTION:

Given chemical strengths between amino acids.
As **chemical link strength is bidirectional**, graph can be considered **UNDIRECTED** and represented as :



In order to proceed further in applying A* search over the graph.
We have to decide between TREE SEARCH and GRAPH SEARCH.

As explained in class:

A* is optimal in TREE SEARCH when the HEURISTIC is ADMISSIBLE.
A* is optimal in GRAPH SEARCH when the HEURISTIC is CONSISTENT.

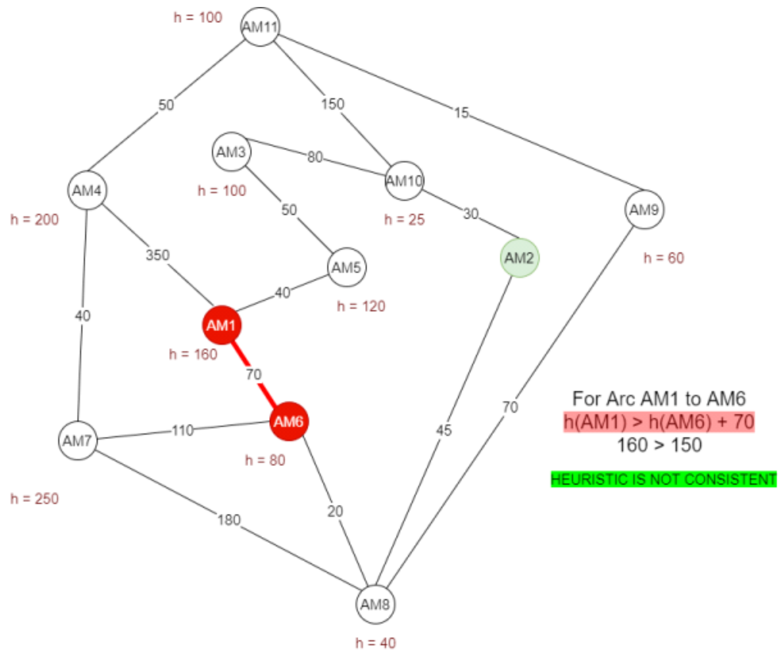
For an admissible but inconsistent heuristic, A* produces a suboptimal solution when GRAPH SEARCH is used.
For an inadmissible heuristic, A* produces suboptimal solution when TREE SEARCH is used.

STEPS TO PROCEED FURTHER:

1. **Check if the heuristic** provided is **consistent**.
2. **If it is consistent** we move forward by using **GRAPH SEARCH**
3. Else we use **TREE SEARCH.**

In order to check consistency for the heuristic provided, for every node we check

$$h(n) \leq c(n, a, n') + h(n')\ [where\ n'\ is\ the\ successor\ of\ node\ n]$$



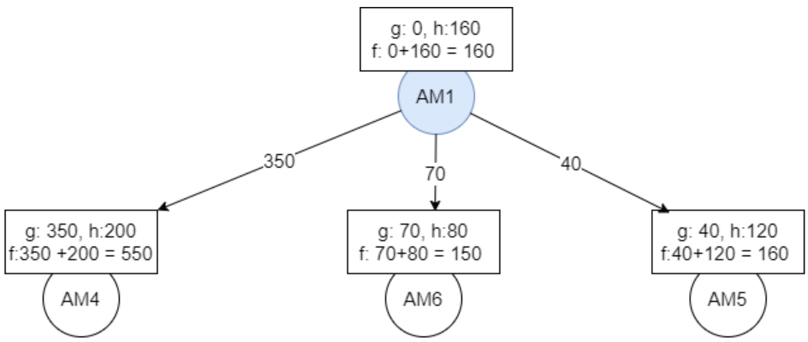
It can also be verified that for arcs between AM11-AM9, AM6-AM8 triangle inequality is not satisfied.

Although the question states :

An oracle also gives you the heuristic distance values to AM2 from each other amino acid in the sample. This heuristic is consistent. Specify if you will use TREE-SEARCH or GRAPH-SEARCH. Motivate your decision. **(5 points)**

Going by the heuristic values provided.
We perform A* using TREE SEARCH on the graph.
As we are performing TREE SEARCH we do not maintain an explored list used to avoid repeated state.

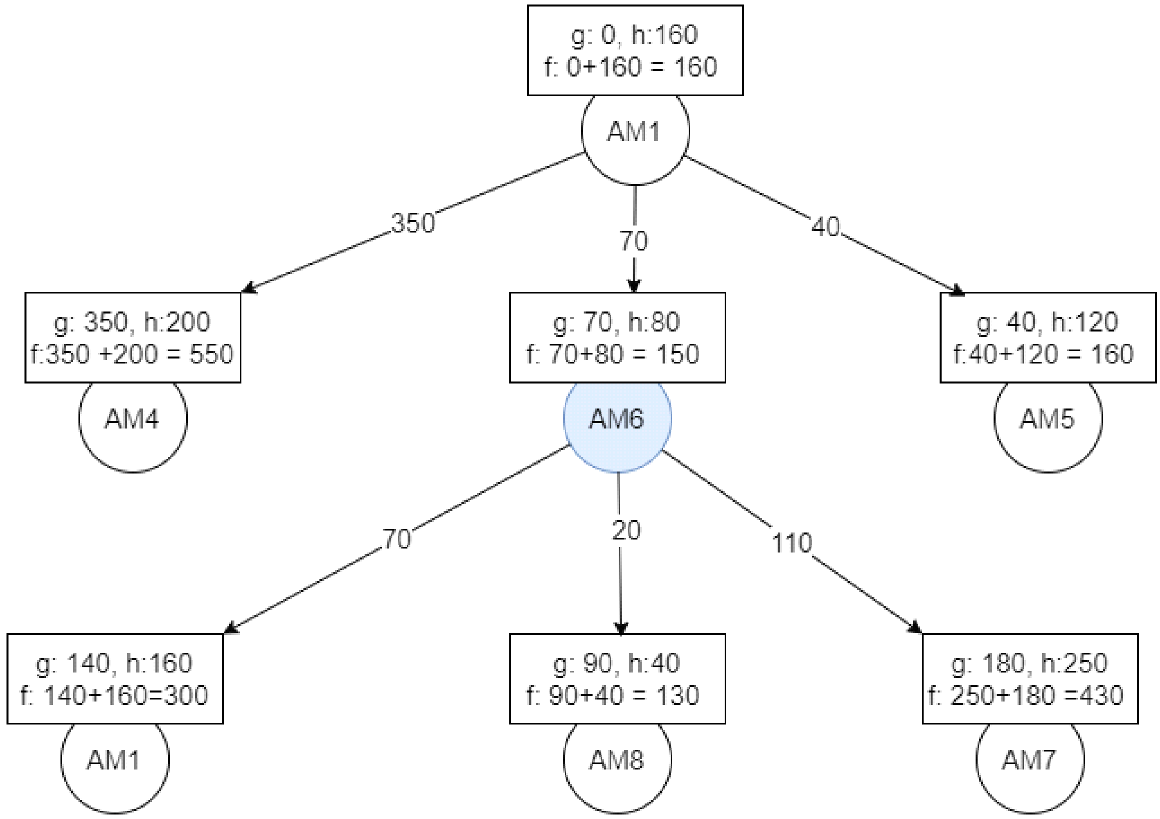
STEP - 1



CURRENT NODE	AM1
CHILDREN	[(AM4, 550), (AM6, 150), (AM5, 160)]
CURRENT PATH	AM1
FRONTIER	[(AM6, 150), (AM5, 160), (AM4, 550)]
NEXT NODE	AM6

Explored list : [(AM1,160)]
{Written only as the question asked, not considered in child expansion strategy}

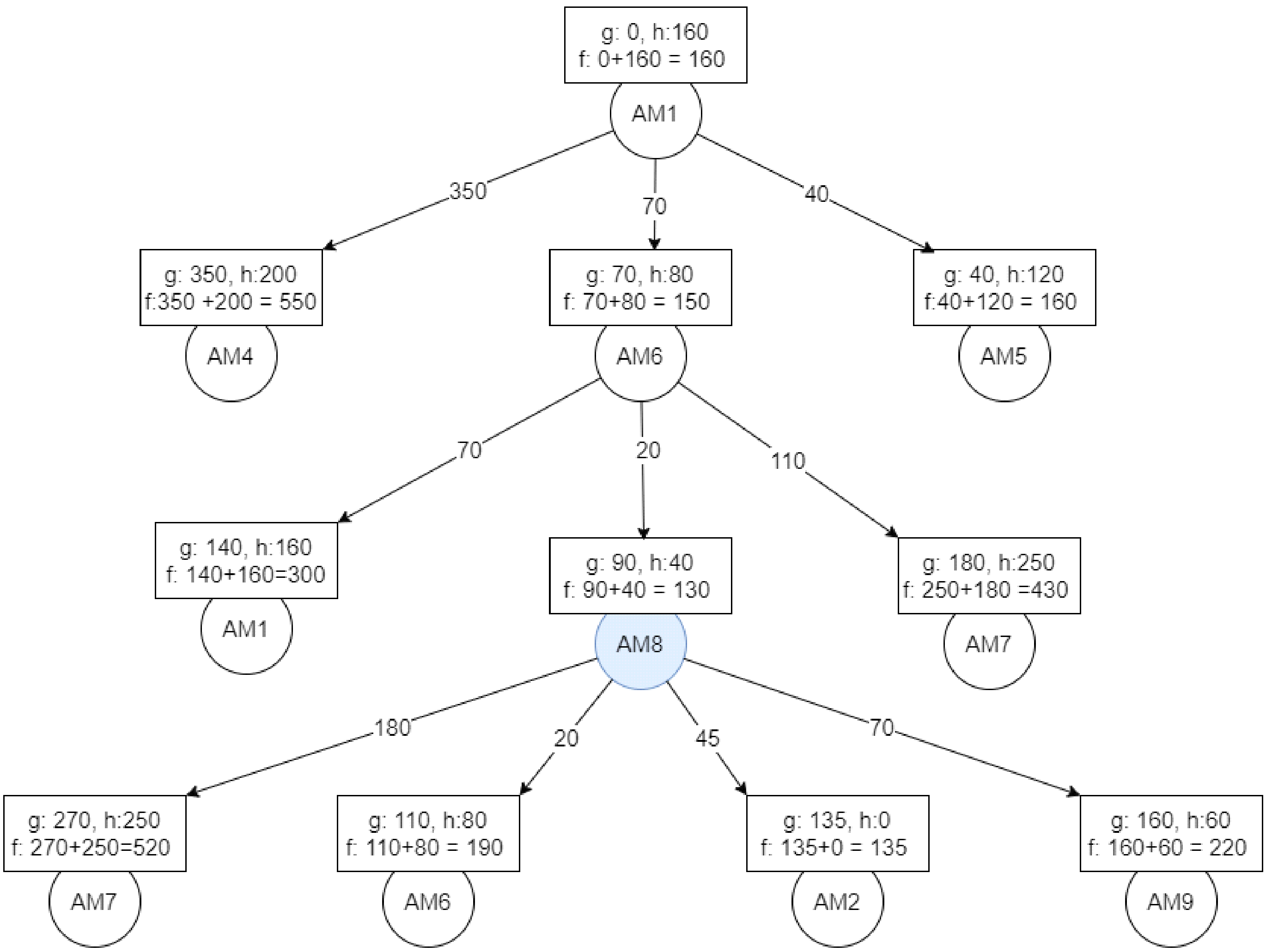
STEP - 2



CURRENT NODE	AM6
CHILDREN	[(AM1, 300), (AM8, 130), (AM7, 430)]
CURRENT PATH	[AM1, AM6]
FRONTIER	[(AM8, 130), (AM5, 160), (AM1, 300), (AM7, 430), (AM4, 550)]
NEXT NODE	AM8

Explored list : [(AM1,160), (AM6, 150)]
{Written only as the question asked, not considered in child expansion strategy}

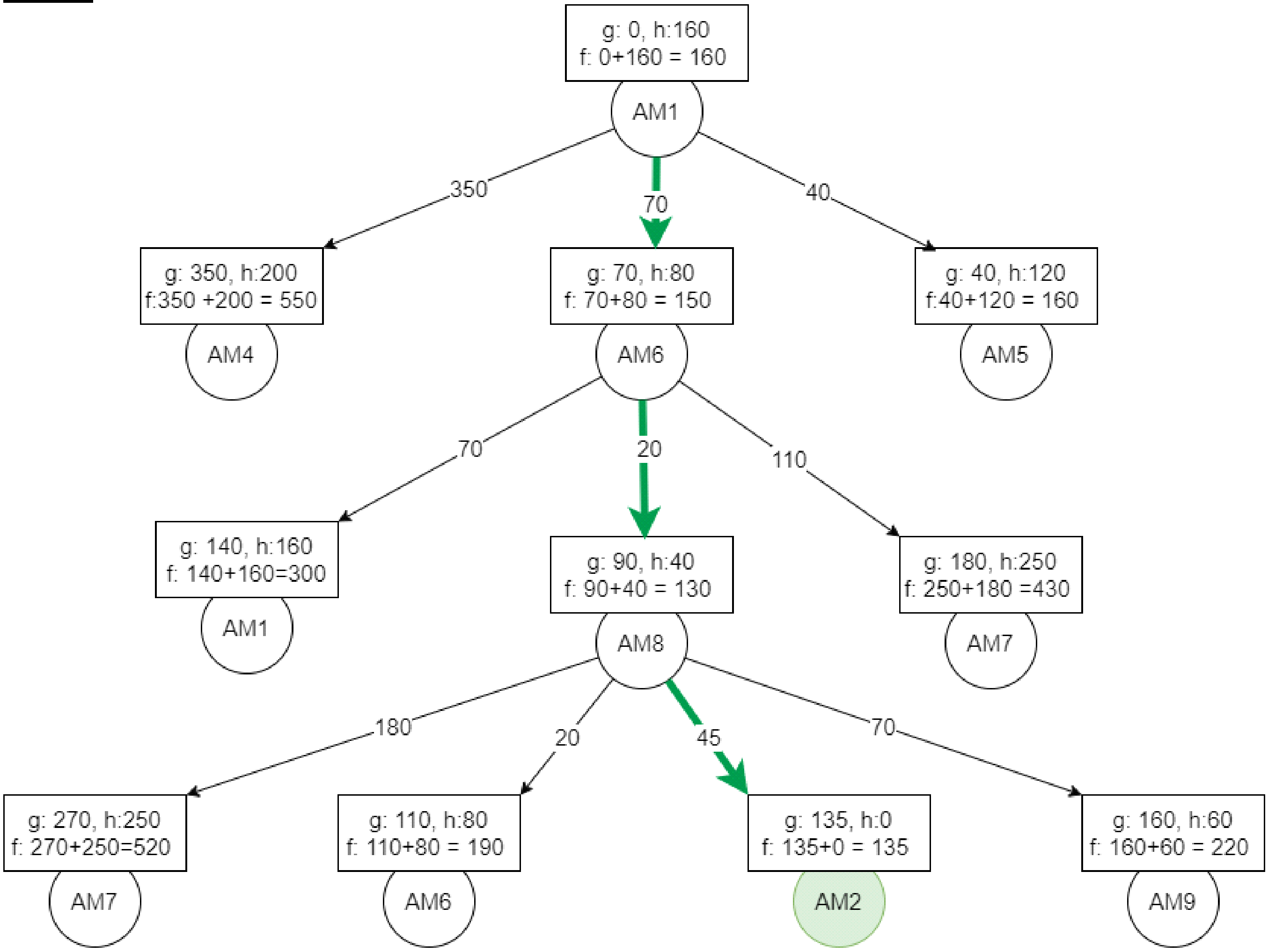
STEP - 3



CURRENT NODE	AM8
CHILDREN	[(AM7, 520), (AM6, 190), (AM2, 135), (AM9, 220)]
CURRENT PATH	[AM1, AM6, AM8]
FRONTIER	[(AM2, 135), (AM5, 160), (AM6, 190), (AM9, 220), (AM1, 300), (AM7, 430), (AM7, 520), (AM4, 550)]
NEXT NODE	AM2

Explored list : [(AM1,160), (AM6, 150), (AM8, 130)]
{Written only as the question asked, not considered in child expansion strategy}

STEP - 4



CURRENT NODE	GOAL STATE (AM2, 135)

SOLUTION PATH :

FATHER (AM2) : AM8
FATHER (AM8) : AM6
FATHER (AM6) : AM1

AM1 ---> AM6 ---> AM8 --> AM2

PATH COST : 135

A more closer observation about the heuristic can result if we perform Uniform Cost Search.
As UCS gives shortest path cost from the start node to the goal node.

Consider AM6 as initial node and AM2 as goal node.

Shortest path from AM6 to AM2 is AM6 --> AM8 --> AM2 with path cost of 20+45 = 65
But for AM6 heuristic value is given as 80.
Overestimating the true cost function.

This is being verified from the above A* search performed :

Provided Heuristic for AM1 is 160
But found path cost 135

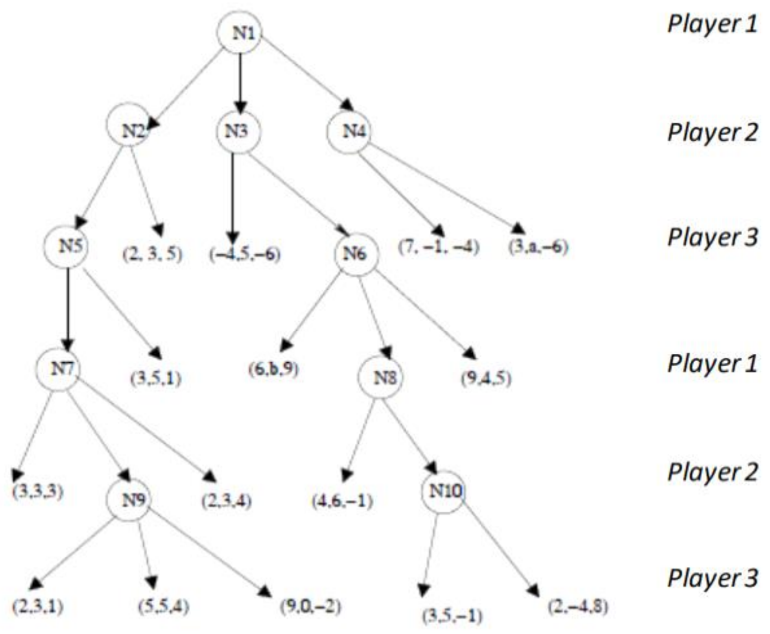
Heuristic overestimated the Path cost.

So, given heuristic is not admissible also.

A* with inconsistent heuristic gives suboptimal result for GRAPH SEARCH
A* with inadmissible heuristic gives suboptimal result for TREE SEARCH.

Problem 2 (50 points)

(a) (**15 points**) Given the following game tree, find the possible values of variables *a* and *b* such that the minimax values in node N1 are (7, −1, −4). Also compute the minimax values at nodes: N2, N3, N4, N5, N6, N7, N8, N9 and N10.



SOLUTION:

Given MINIMAX value for Node N1 as (7, -1, -4).
Depth first search is performed in evaluation of Minimax values of Nodes.

Considering utility values as tuples of (x, y, z) :
P1 : Maximizes his utility by maximizing x
P2 : Maximizes his utility by maximizing y
P3: Maximizes his utility by maximizing z

NODE	N9
MINIMAX	Max_{Y}[(2,3,1), (5,5,4), (9, 0,-2)] = (5,5,4)
PLAYER	P2

NODE	N7
MINIMAX	Max_{X}[(3,3,3), (5,5,4), (2, 3,4)] = (5,5,4)
PLAYER	P1

NODE	N5
MINIMAX	Max_{Z}[(5,5,4), (3, 5, 1)] = (5,5,4)
PLAYER	P3

NODE	N2
MINIMAX	Max_{Y}[(5,5,4), (2, 3, 5)] = (5,5,4)
PLAYER	P2

NODE	N10
MINIMAX	Max_Y[(3,5,-1), (2,-4,8)] = (3,5,-1)
PLAYER	P2

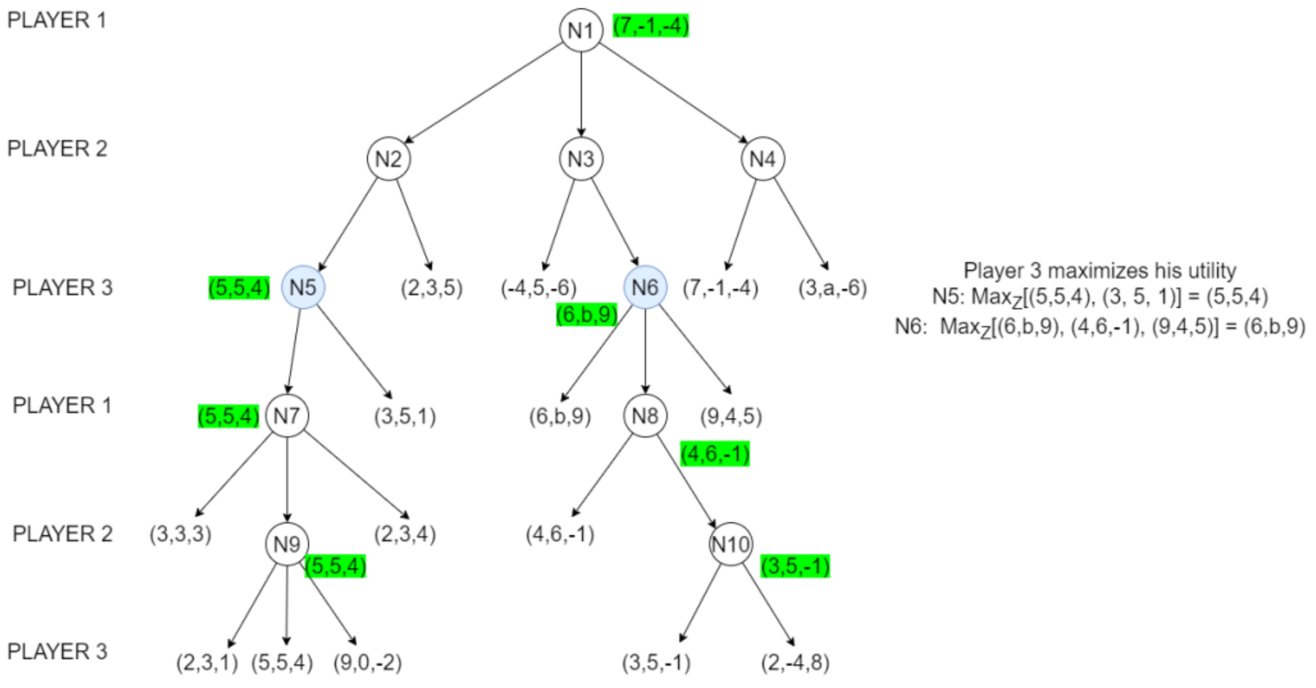
NODE	N8
MINIMAX	Max_X[(4,6,-1), (3,5,-1)] = (4,6,-1)
PLAYER	P2

NODE	N6
MINIMAX	Max_Z[(6,b,9), (4,6,-1), (9,4,5)] = (6,b,9)
PLAYER	P2

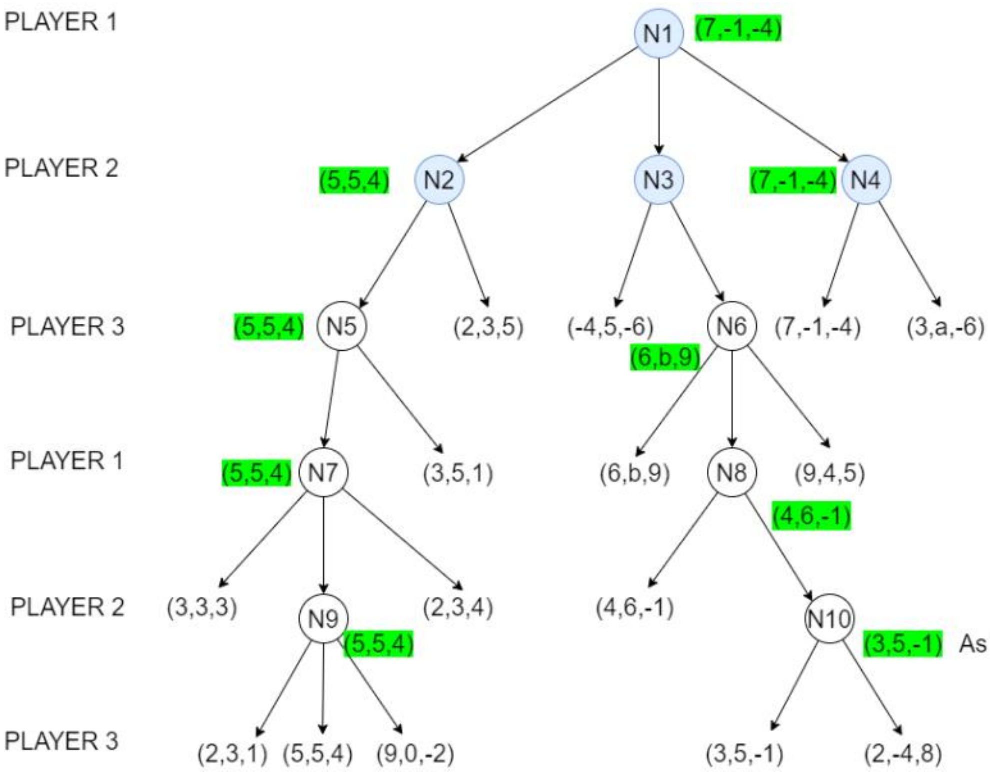
NODE	N3
MINIMAX	Max_Y[(-4, 5, -6), (6,b,9)]
PLAYER	P2

NODE	N4
MINIMAX	Max_Y[(7, -1, -4), (3,a,-6)]
PLAYER	P2

Considering the game tree after filling up with the MINIMAX values computed so far we get :



Final game tree with results is as follows :



Player 2 maximizes his utility

N2: Max_Y[(5,5,4), (2, 3, 5)] = (5,5,4)

N3: Max_Y[(-4, 5, -6), (6,b,9)]

N4: Max_Y[(7, -1, -4), (3,a,-6)]

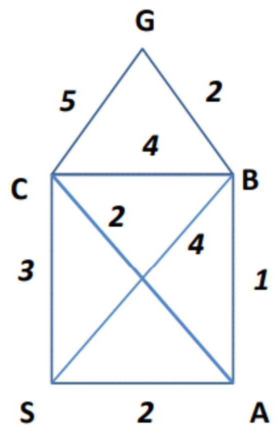
As (7,-1,-4) is only present in the branch of N4
Inorder to obtain (7,-1,-4) at N1,
value at N4 = (7,-1,-4)

For N4 to have a value of (7,-1,-4) satisfying
N4: Max_Y[(7, -1, -4), (3,a,-6)]

$a < -1$

Considering options at N3,
Possible values for Player 1 are (-4,6)
-4 from (-4, 5, -6) and 6 from (6,b,9)
As both -4,6 are less than 7 value at N3 can be anything
So, b can be anything

(b) (**15 points**) An agent stating in state *S* should reach the goal state *G*. If the possible states the agent can reach are *A, B, C* or *G*, as depicted bellow:



And as shown in the figure: cost(S→A) = 2; cost(S→B) = 4; cost(S→C) = 3; cost(A→B) = 1; cost(A→C) = 2; cost(B→C) = 4; cost(B→G) = 2; cost(C→G) = 5; you are asked to:

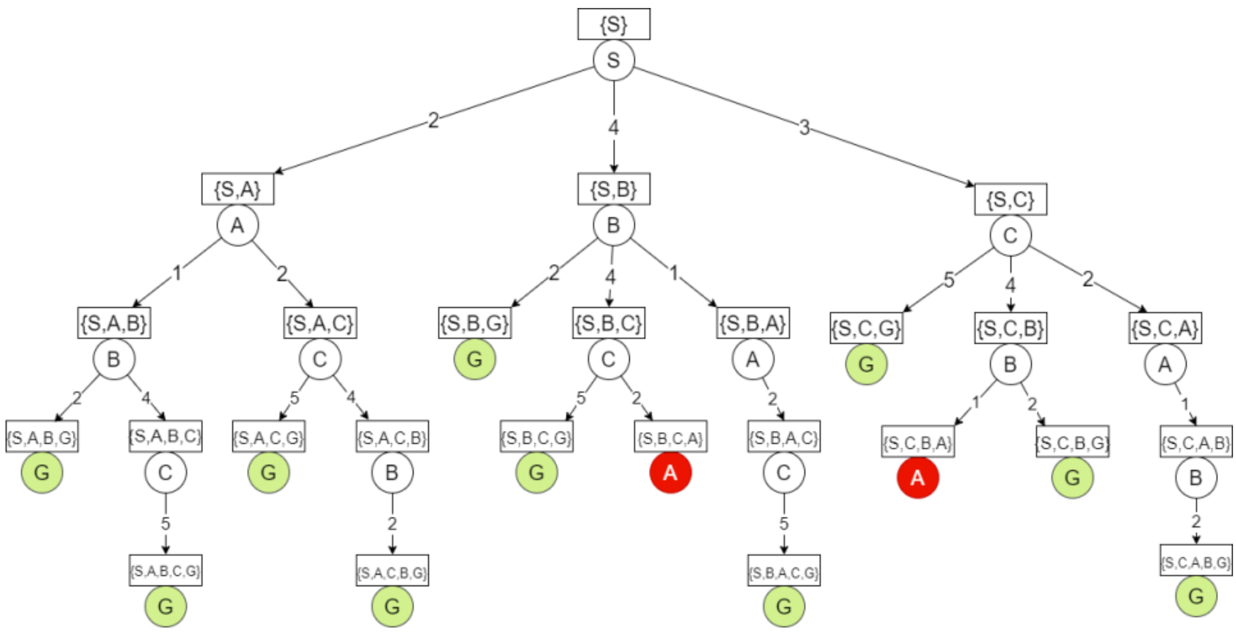
(a) draw the search tree that allows the agent to travel from *S* to *G*, knowing that the agent *cannot ever visit S again*, and cannot visit any state more than once. Show in the search tree all the ways in which the agent can get from the state *S* to the goal state *G*; (**5 points**) How many ways of getting to the goal state *G* from *S* are there? (**2 points**) *HINT:* Any solution path starts in *S* and ends in *G* but does not have to visit all other nodes! However, it cannot visit more than one any node!

(b) What is the least costly and the costliest way for the agent to get from state *S* to state *G*? Show the least costly path (**2 points**) and specify how much it costs (**2 points**). Show the costliest path (**2 points**) and specify how much it costs (**2 points**) and show how you have computed the costs.

SOLUTION:

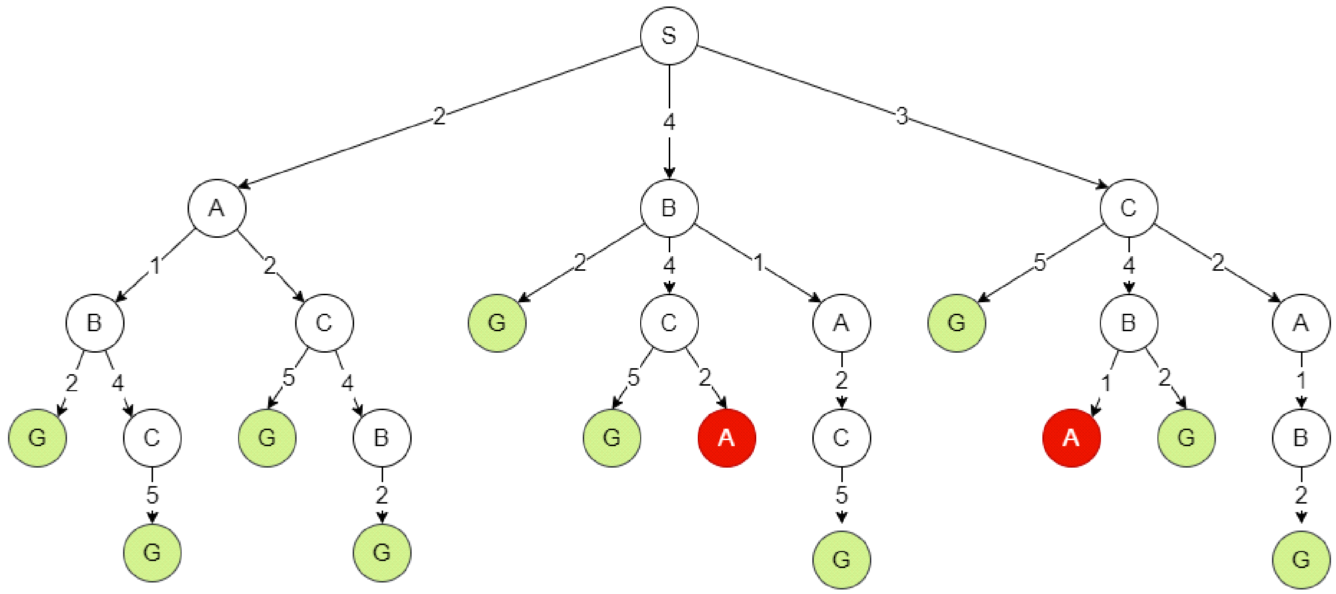
Assumptions

1. Considering the links between nodes as **BIDIRECTIONAL**
2. For any node in search tree, along the path from the node to the root it should not have repeated states. [Without cycles]

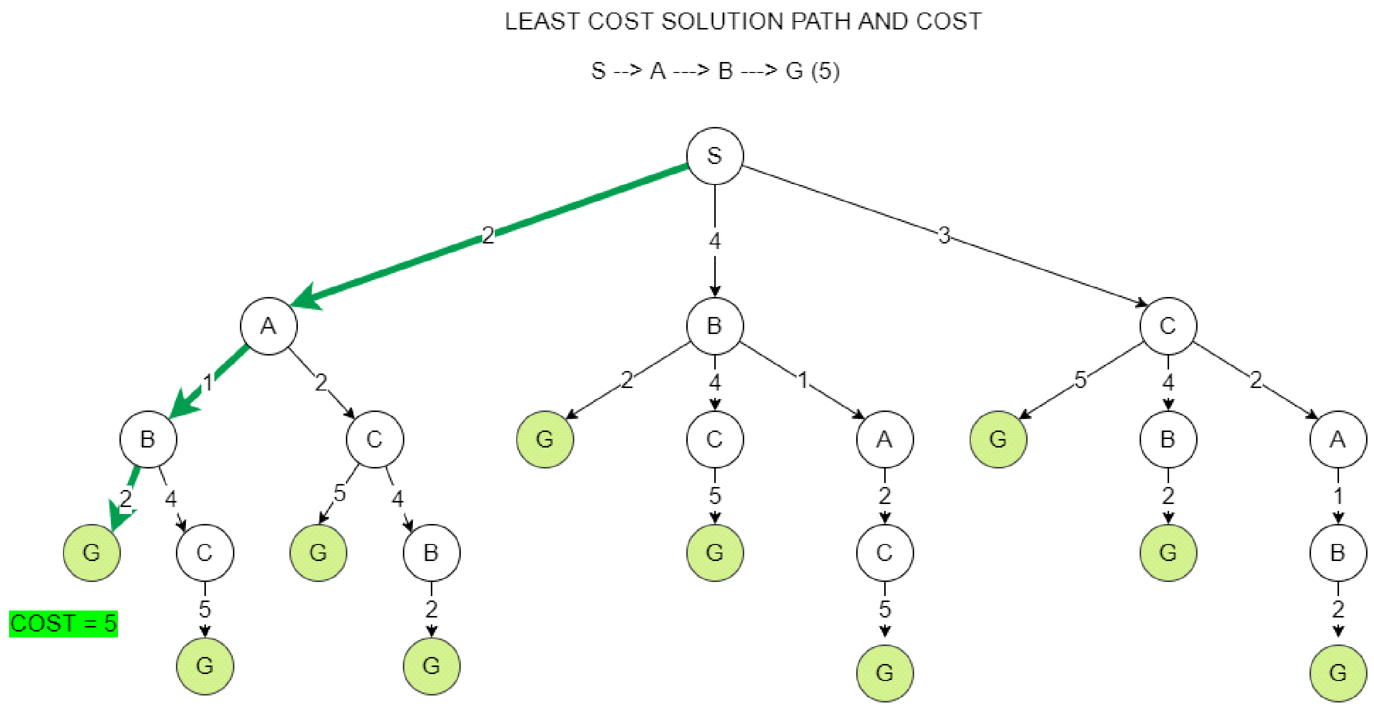


Complete search tree can be considered as follows:

Number of ways of getting to goal state = Number of goal states as leaf nodes = 10



Least cost path is shown as follows:



This can be computed using Uniform Cost Search.
Best way to reach the goal in the shortest distance

Applying UCS over the above tree to find the least cost path

1) Current Node: S

Children: A(2), C(3), B(4)

Frontier: A(2), C(3), B(4)

Next node: A

2) Current Node: A

Children: B(3,A), C(4,A)

Frontier: C(3,S), B(3,A)

B(4,S), C(4,A)

Next node: C(3,S)

3) Current Node: C(3,S)

Children: G(8,C), B(7,C), A(5,C)

Frontier: B(3,A), B(4,S), C(4,A)

A(5,C), B(7,C), G(8,C)

Next node: B(3,A)

4) Current Node: B(3,A)

Children: G(5,B), C(7,B)

Frontier:

B(4,S), C(4,A)

A(5,C), G(5,B), C(7,B)

B(7,C), G(8,C)

Next node: B(4,S)

5) Current Node: B(4,S)

Children: G(6,B), C(8,B), A(5,B)

Frontier: C(4,A), A(5,C), G(5,B), A(5,B)

G(6,B), C(7,B), B(7,C), G(8,C), C(8,B)

Next node: C(4,A).

6> Current Node : C (4, A)

Children : G (9, C), B(8, C)

Frontier : A(5, C), G(5, B), A(5, B)

G(6, B), C(7, B), B(7, C), G(8, C), C(8, B), B(8, C)

G(9, C),

Next node : G (5, B)

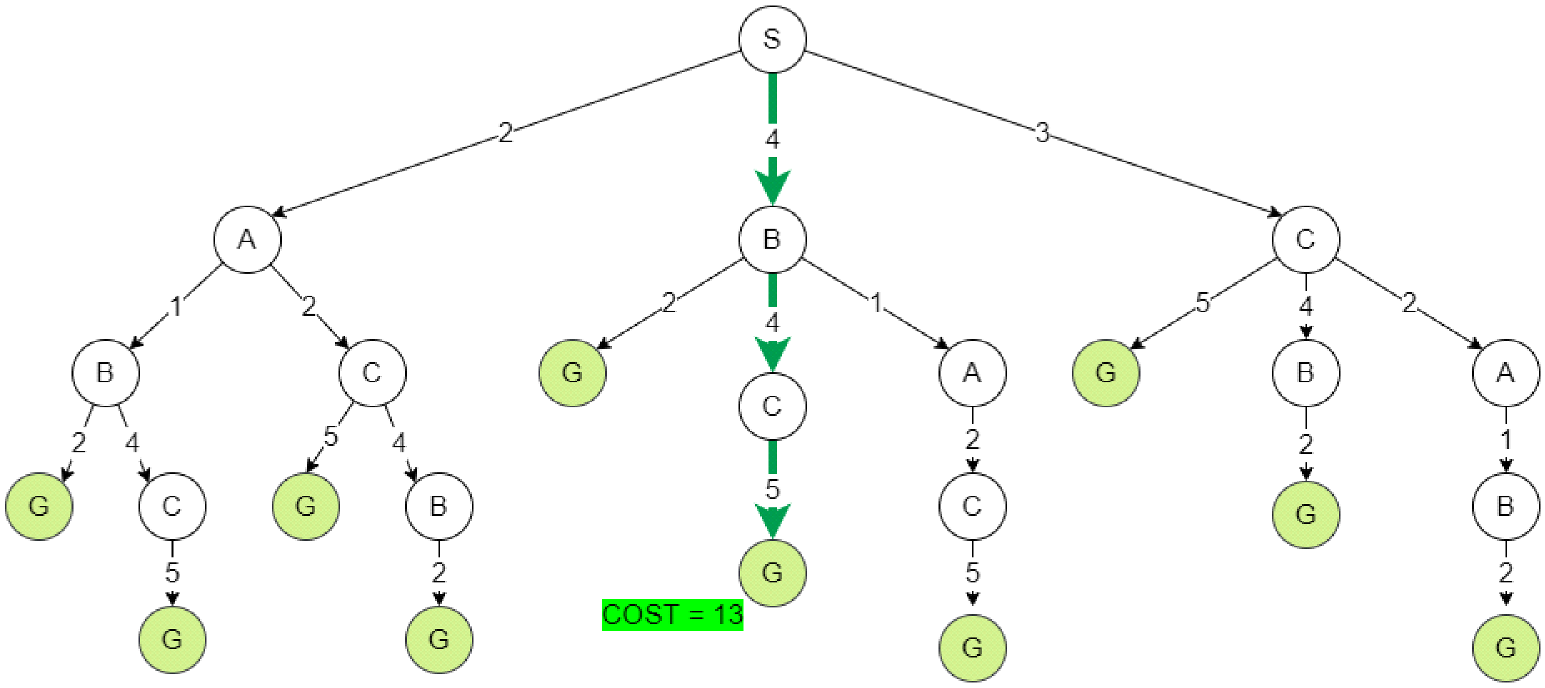
7> Current Node : G (5, B)

$\boxed{\text{GOAL}}$

Path: S → A → B → G [5]

MAXIMUM COST SOLUTION PATH AND COST

S --> B --> C --> G (13)



Maximum cost path in the tree can also be computed with Uniform Cost Search by considering path costs negative I,e

1) Current Node: S (0)

Children: A (-2, S), B (-4, S), C (-3, S)

Frontier: B (-4, S), C (-3, S), A (-2, S)

Next node: B (-4, S)

2) Current Node: B (-4, S)

Children: G (-6, B), C (-8, B), A (-5, B)

Frontier: C (-8, B), G (-6, B), A (-5, B), C (-3, S), A (-2, S)

Next node: C (-8, B)

3) Current Node: C (-8, B)

Children: G (-13, C)

Frontier: G (-13, C),
C (-8, B), G (-6, B), A (-5, B), C (-3, S), A (-2, S)

Next node: G (-13, C)

4) Current Node: G (-13, C)

Goal state ⇒ Path: S → B → C → G [-13 ⇒ 13]

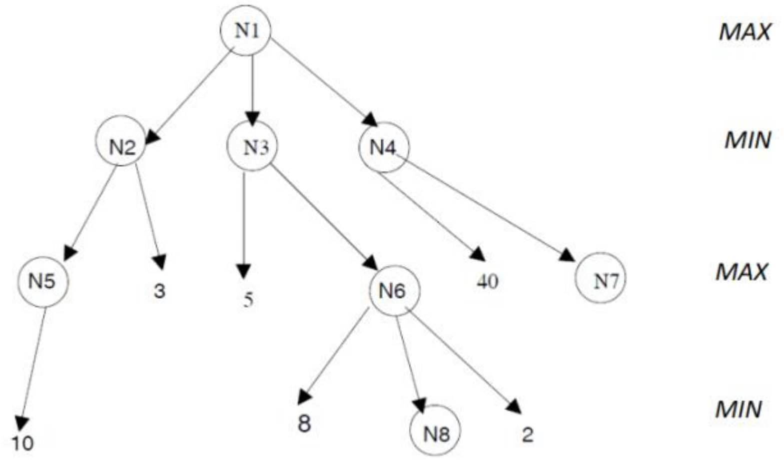
(c) (**20 points**) Given the game tree below, compute the value of alpha and beta at following nodes, if the order is the same as in depth-first search:

(1) alpha and beta at node N3 before and after visiting the terminal node with utility 5. Also show the values of alpha and beta in N3 after visiting N6 (Hint: Show also the values of alpha and beta at all nodes visited before you reached N3.); **(3 points)**

(2) alpha and beta at node N6; **(2 points)**

(3) alpha and beta at node N1 after visiting N7, if the node N7 has a child node with an utility value *x* (after you visited all nodes illustrated in the Figure) **(5 points)**

(4) should the game tree be pruned? If yes, how? **(10 points)**



SOLUTION:

NODE	α	β	VALUE FUNCTION
N1	-∞	∞	max-value : -∞
N2	-∞	∞	min-value : ∞
3	return 3		
N2	-∞	3	min-value : 3
N5	-∞	3	max-value : -∞
10	return 10		
N5	-∞	3	max-value : 10 max-value >= β β - CUT if there were any other children
N2	-∞	3	min-value : 3
N1	3	∞	max-value : 3
N3	**3**	**∞**	**min-value : ∞**
5	return 5		
N3	**3**	**5**	**min-value : 5**
N6	3	5	max-value : -∞
8	return 8		
N6	**3**	**5**	**max-value : 8** **max-value >= β** **β-CUT [Prunes Nodes N8 and 2]**

N3	3	5	min-value : 5
N1	5	∞	max-value : 5
N4	5	∞	min-value : ∞
40	return 40		
N4	5	40	min-value : 40
N7	5	40	max-value : -∞
X	**return x**		

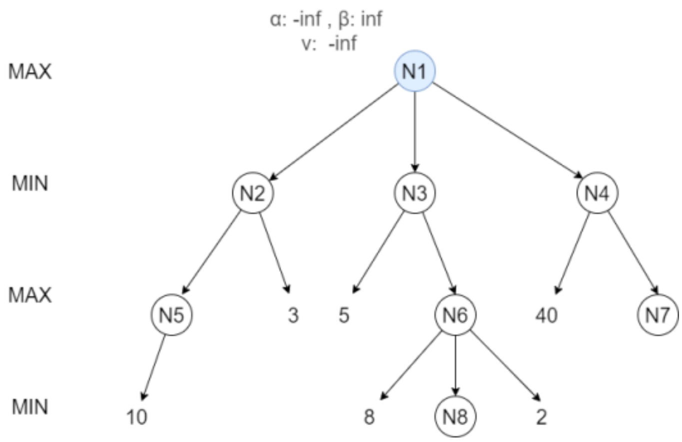
N7	if x ≥ 40 :
	α = 5 [remains same, due to β-CUT]
	else if 5<x<40:
	α = x
	else:
	α=5
		40	max-value : x
			if x ≥ 40 :
			max-value >= β
			[β-CUT of any other children of N7]
			else:
			max-value = x

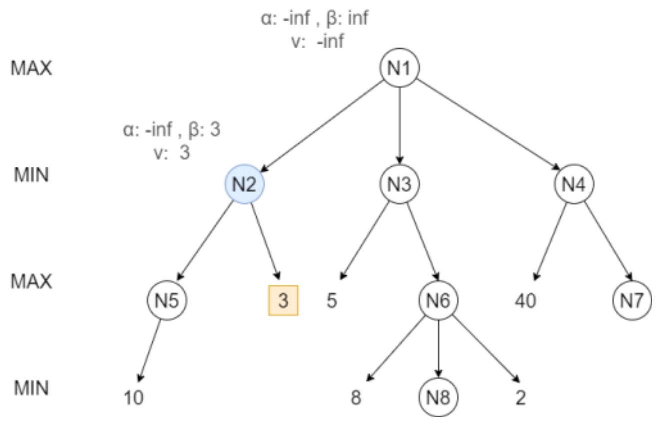
N4	5	if x ≤ 5:
		β = 40 [remains same due to α-CUT]
		else if 5 < x ≤ 40:
		β = x
		else:
		β = 40 [as min-value of node remains same]
			min-value : min(40, x)
			if x < 5 :
			min-value = x
			[α-CUT prunes other children of N4]
			else if 5 < x ≤ 40:
			min-value = x
			else:
			min-value = 40

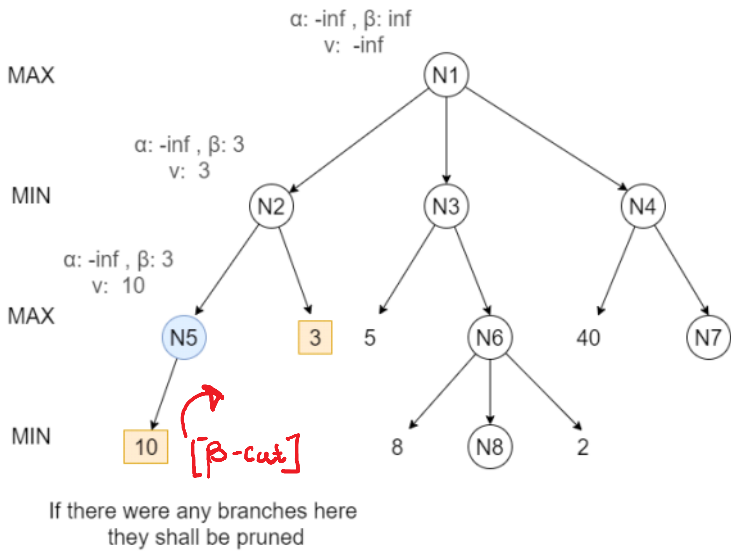
N1	**if x ≤ 5:** **α = 5 [as max-value remains same]** **else if 5 < x ≤ 40:** **α = x** **else:** **α=40**	**∞**	**max-value : max(5, min(40,x))** **if x < 5:** **max-value = 5** **else if 5 < x ≤ 40:** **max-value = x** **else:** **max-value = 40**

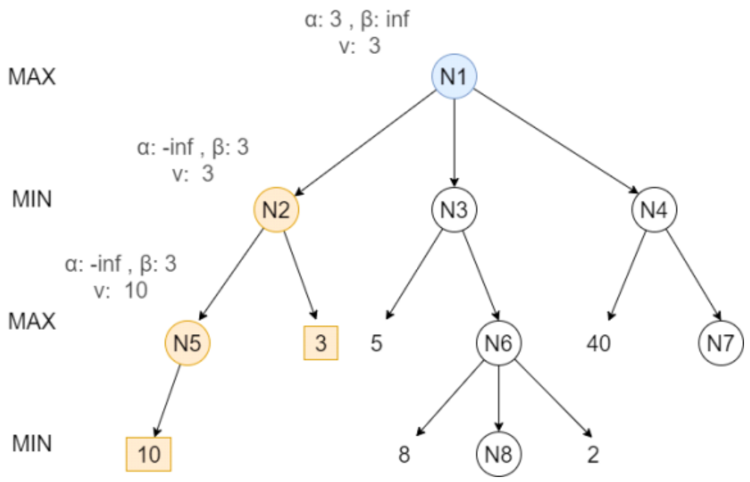
N1 has MINIMAX as max-value corresponding to the table

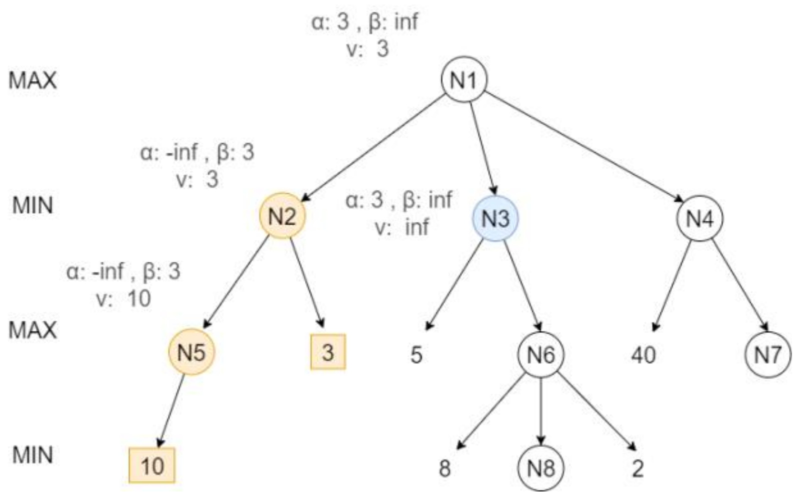
Detailed trace of the alpha beta procedure at few points is listed below indicating pruned nodes

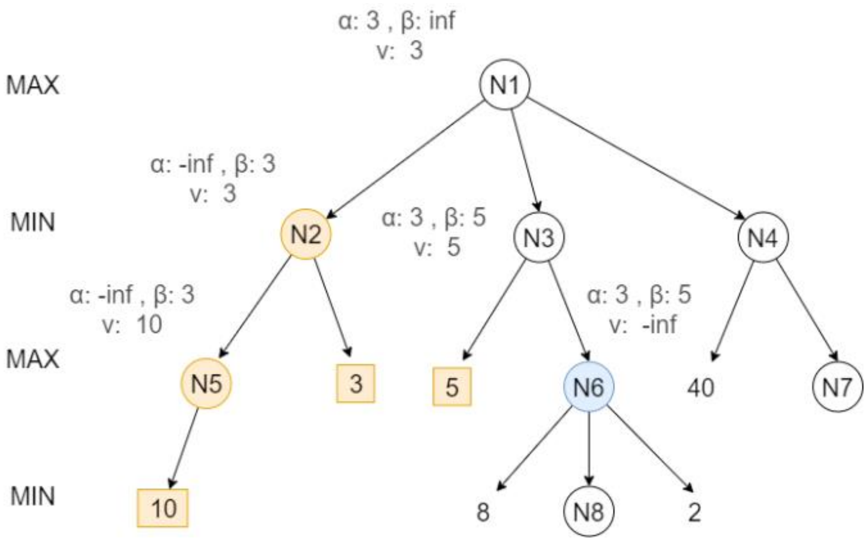


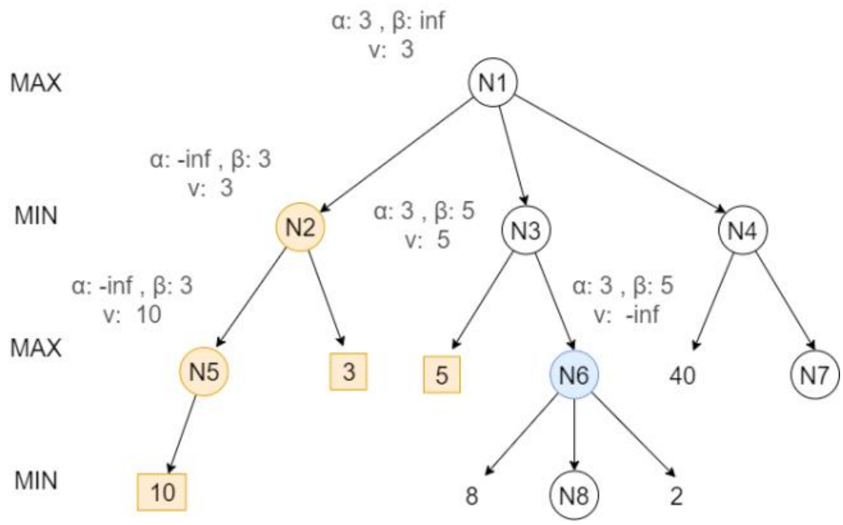


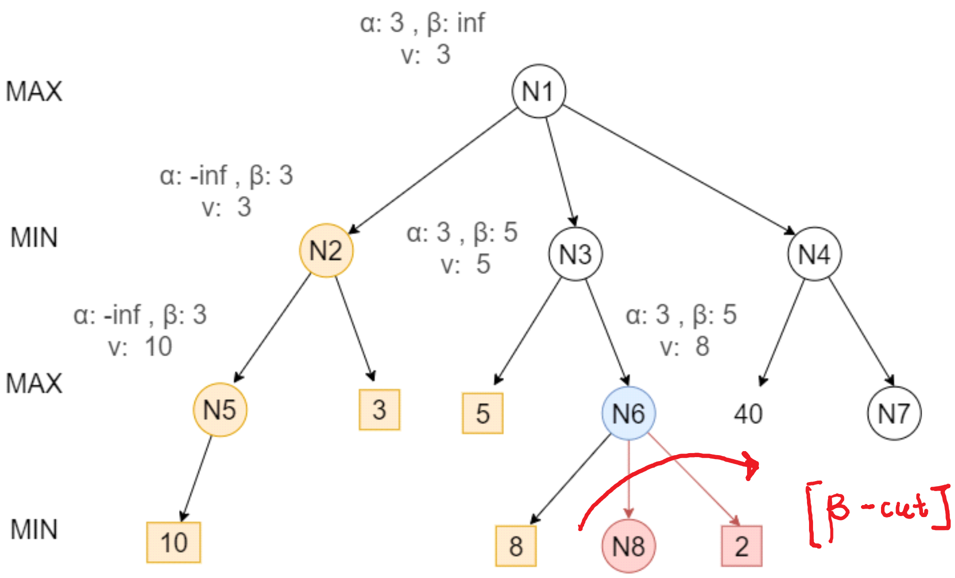

α: -inf , β: inf
v: -inf
MAX
N1
α: -inf , β: 3
v: 3
MIN
N2
N3
N4
α: -inf , β: 3
v: 10
MAX
N5
3
5
N6
40
N7
MIN
10
[β-cut]
8
N8
2
If there were any branches here
they shall be pruned

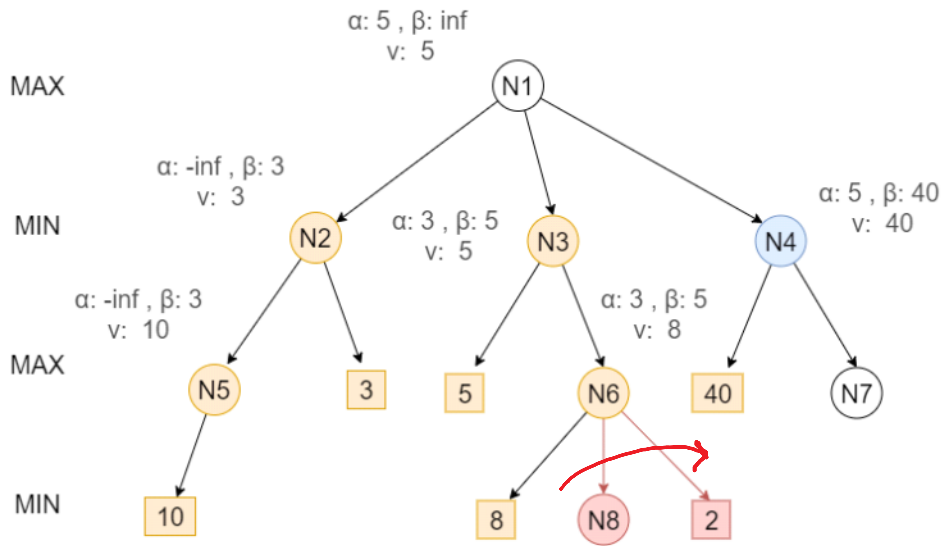

α: 3 , β: inf
v: 3
MAX
N1
α: -inf , β: 3
v: 3
MIN
N2
N3
N4
α: -inf , β: 3
v: 10
MAX
N5
3
5
N6
40
N7
MIN
10
8
N8
2

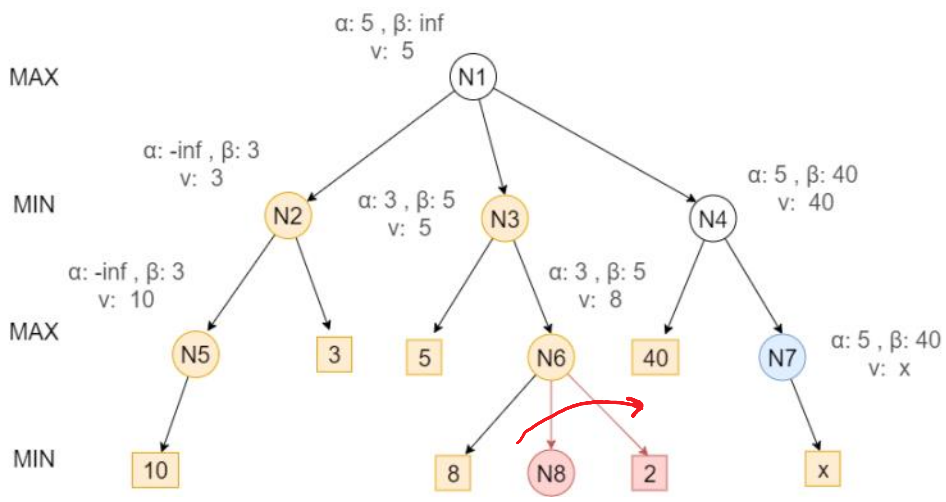

α: 3 , β: inf
v: 3
MAX
N1
α: -inf , β: 3
v: 3
MIN
N2
α: 3 , β: inf
v: inf
N3
N4
α: -inf , β: 3
v: 10
MAX
N5
3
5
N6
40
N7
MIN
10
8
N8
2


α: 3 , β: inf
v: 3
MAX
N1
α: -inf , β: 3
v: 3
MIN
N2
α: 3 , β: 5
v: 5
N3
N4
α: -inf , β: 3
v: 10
α: 3 , β: 5
v: -inf
MAX
N5
3
5
N6
40
N7
MIN
10
8
N8
2


α: 3 , β: inf
v: 3
MAX
N1
α: -inf , β: 3
v: 3
MIN
N2
α: 3 , β: 5
v: 5
N3
N4
α: -inf , β: 3
v: 10
α: 3 , β: 5
v: -inf
MAX
N5
3
5
N6
40
N7
MIN
10
8
N8
2


α: 3 , β: inf
v: 3
MAX
N1
α: -inf , β: 3
v: 3
MIN
N2
α: 3 , β: 5
v: 5
N3
N4
α: -inf , β: 3
v: 10
α: 3 , β: 5
v: 8
MAX
N5
3
5
N6
40
N7
MIN
10
8
N8
2
[β-cut]





Different possibilities arise for different values of $\underline{x}$

As shown in the <u>table</u>