Birzeit University

Department of Electrical and Computer Engineering

Second Semester, 2021-2022

Artificial Intelligence - ENCS434

Midterm Exam, May 10, 2022

Time Allowed: 75 Minutes.

Name: ....Sample Solution..........ID: ....

Section -1- Dr. Adnan Yahya **Section -2- Dr. Adnan Yahya**

Section -3- Yazan Abu Farha **Section -4- Yazan Abu Farha**

Please answer All Questions.

Question #	SOC	Max Grade	Achieved Grade
1		20	
2		24	
3		23	
4		15	
5		25	
Total		105	

Qustion#1 [20] Resolution Theorem Proving. Your Knowledge Base (KB) contains the following statements: You are asked to prove the goal statement (¬ F). After converting your KB to CNF and adjoining the negated goal statement you have (in clausal form):

1- (A→)
2- (B→A)
3- (C→A)
4- (D→B)
5- (E→B)
6- (F→C v D v E)

Convert the KB and Goal negation into clausal form.

1- (A→): ¬A
2- (B→A) : ¬B v A
3- (C→A) : ¬C v A
4- (D→B) : ¬D v B
5- (E→B) : ¬E v B
6- (F→C v D v E) : ¬F v C v D v E
7- Goal negation:F

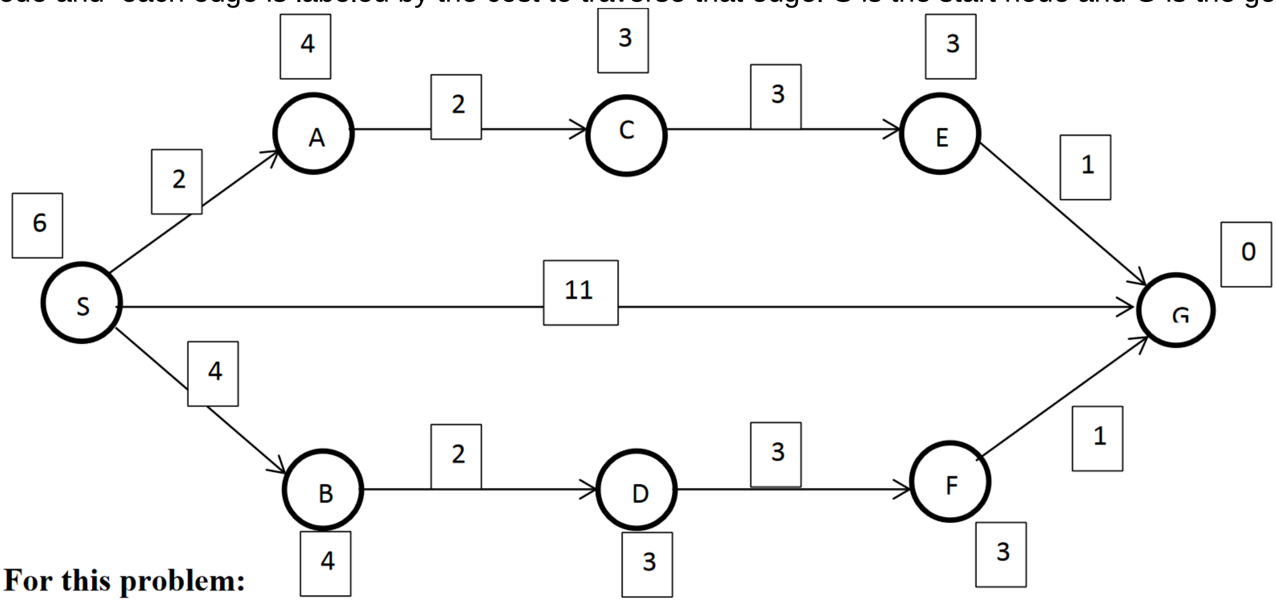
Produce a refutation resolution proof that ¬ F (NOT F) is true.

Steps: which clauses resolve to give which clause with a new number.

1- 1+2→8: ¬B
2- 1+3→9: ¬C
3- 4+8 → 10: ¬D
4- 5+8 → 11: ¬E
6- 6+9→ 12: ¬F v D v E
7- 10+12→ 13: ¬F v E
8- 11+13→ 14: ¬F
9- 7+14→ empty clause

<u>Qustion#2 [24]</u>

Given the graph below, each node is labeled by a capital letter and the value of a heuristic function at that node and each edge is labeled by the cost to traverse that edge. S is the start node and G is the goal node.



For this problem:

Perform the each of the 3 Search algorithms on this graph, filling in the table below. Indicate the values of each node on the queue as shown in the first two rows of the table for A*. You should not need all the lines provided in the table.

- Write the priority queue as <node-label>**=**<f-value>
- The first one for A* is done for you, as an example.

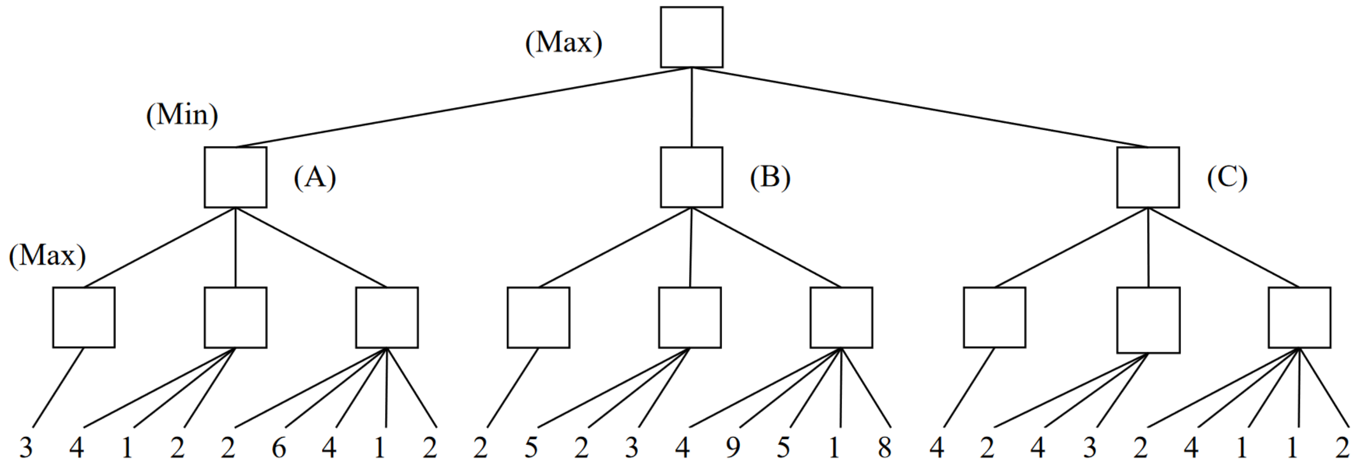
	A* Algorithm		**Greedy Algorithm**		**Uniform Cost Algorithm**	
Iteration	Node Expanded	Priority Queue	Node Expanded	Priority Queue	Node Expanded	Priority Queue
0		S=6		S=6		S=6
1	S=6	A=6; B=8; G=11	S=6	A=6; B=4; G=0	S=6	A=2; B=4; G=11
2	A=6	B=8; C=7; G=11	G=0	A=6; B=4;	A=2	C=4; B=4; G=11
3	C=7	B=8; E=10; G=11			B=4	C=4; D=6; G=11
4	B=8	D=9; E=10; G=11			C=4	E=7; D=6; G=11
5	D=9	E=10; F=12; G=11			D=6	E=7; F=9; G=11
6	E=10	F=12; G=8; G=11			E=7	G=8; F=9; G=11
7	G=8	F=12; G=11			G=8	-----

write down the order in which the states are visited by the following search algorithms. If a state is visited more than once, write it each time. Ties (e.g., which child to first explore in depth-first search) should be resolved according to **alphabetic order** (i.e. prefer A before Z). Remember to include the start and goal states in your answer. Assume that algorithms execute the goal check when nodes are visited, not when their parent is expanded to create them as children.

Qustion#3 [23]: Adversarial Search. Given the search tree shown below….

a) 7% Fill in the squares and circles with the backed-up values resulting from a regular minimax search (no Alpha-Beta pruning.

b) 3% What is the Root value and the move taken (mark the branch with "Move taken"). **4 and Root to C.**

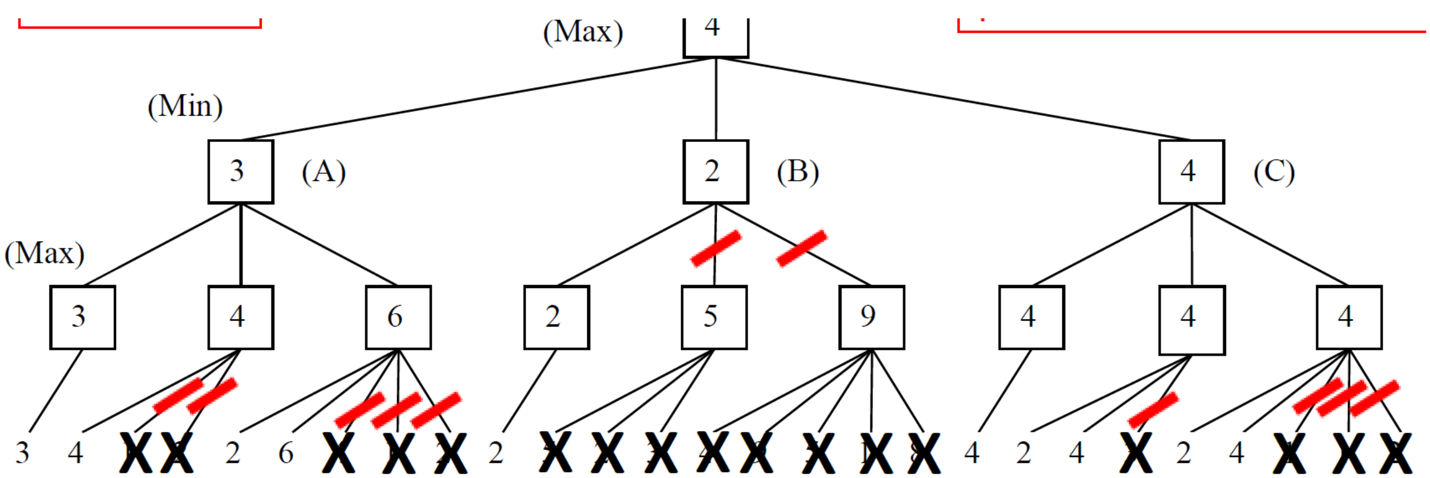


4

A: 3 **B: 2** **C: 4**

3 **3** **6** **2** **5** **9** **4** **4** **4**

c)13% Show how a depth-first search with alpha-beta cutoffs would work, indicating all α and β cut-offs by drawing an X through the unexplored branches. Assume that children are explored from left to right.



Qustion#4 [15]: For the following statements, answer by marking with X [**True or False**] and justify your answer using **one line of text**.

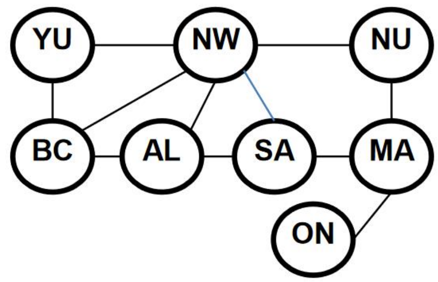
1. □**True** □**False** Breadth first search always expands less nodes than depth first search.
2. □**True** □**False** The pruning in min-max improves efficiency but can cause the loss of some solutions.
3. □**True** □**False** Iterative deepening search is guaranteed to expand more nodes than breadth first search (on any graph whose root is not the goal).
4. □**True** □**False** A formula F (sentence) is unsatisfiable if it has no models.
5. □**True** □**False** An agent can change its environment using actuators.
6. □**True** □**False** Random restarts are used in local search to diminish the problem of local maxima.
7. □**True** □**False** Doubling your computer's speed allows you to double the depth of a tree search given the same amount of time
8. □**True** □**False** Backtracking search on CSPs, while generally much faster than general purpose search algorithms like A*, still requires exponential time in the worst case.
9. □**True** □**False** One reason to use forward checking in a CSP problem is in order to detect failures quickly and backtrack earlier.
10. □**True** □**False** For a search problem, the path returned by uniform cost search may change if we add a positive constant C to every step cost.
11. □**True** □**False** If h1(n) and h2(n) are two admissible A* heuristics, then their average h3(n) = ½ h1(n) + ½ h2(n) must also be admissible.
12. □**True** □**False** One way that forward checking can speed CSP-solving is by detecting dead-end choices in backtracking search.
13. □**True** □**False** Algorithm A* search using the heuristic h(n) = c for some fixed constant c > 0 is guaranteed to find an optimal solution.
14. □**True** □**False** If h1 and h2 are both admissible heuristics, it is always better to use the heuristic h3(n) = max(h1(n), h2(n)) rather than the heuristic h4(n) = min(h1(n), h2(n)).
15. □**True** □**False** Minimax can be applied to three player games, for example by having each player trying to maximize own chances of winning.

Please transfer your answers into the following table (if not -3 points).

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
□True □False	□True □False	□True □False	□True □False	□True □False	□True □False	□True □False	□True □False	□True □False	□True □False	□True □False	□True □False	□True □False	□True □False	□True □False

Qustion#5 [25] CSPs

Consider a CSPs for map coloring of Canada. Each region on the map is a variable, and their values are chosen from {R, G, B}. Adjacent regions **cannot** have the same color. The figures below show the constraint graph. Please answer the following:



AL = Alberta
BC = British Columbia
MA = Manitoba
NW = Northwest Territories
NU = Nunavut
ON = Ontario
SA = Saskatchewan
YU = Yukon Territory

1. NW has been assigned value B, as shown. Cross out all values that would be eliminated by Forward Checking (FC):

AL	**BC**	**MA**	**NW**	**NU**	**ON**	**SA**	**YU**
RG~~B~~	RG~~B~~	RGB	B	RG~~B~~	RGB	RG~~B~~	RG~~B~~

2. NW has been assigned B and AL has been assigned R, as shown; but no constraint propagation has been done. Cross out all values that would be eliminated by Arc Consistency (AC-3 in the slides).

AL	**BC**	**MA**	**NW**	**NU**	**ON**	**SA**	**YU**
R	~~R~~G~~B~~	R~~G~~B	B	RG~~B~~	RGB	~~R~~G~~B~~	R~~GB~~

3. Consider the assignment below. AL has been assigned B and constraint propagation has been done, as shown. List all unassigned variables (in alphabetical order) that might be selected now by the Minimum-Remaining-Values (MRV) Heuristic:_ **NW, BC, SA**________

AL	**BC**	**MA**	**NW**	**NU**	**ON**	**SA**	**YU**
B	RG	RGB	RG	RGB	RGB	RG	RGB

4. Consider the assignment below. (It is the same assignment as in problem 3c above.) AL has been assigned B and constraint propagation has been done, as shown. Ignoring the MRV heuristic, list all unassigned variables (in alphabetical order) that might be selected now by the Degree or Most Constraining Variable Heuristic (DH, MCV): NW _____________

AL	**BC**	**MA**	**NW**	**NU**	**ON**	**SA**	**YU**
B	RG	RGB	RG	RGB	RGB	RG	RGB

5. Consider the assignment below. (It is the same assignment as in problem 3c above.) AL has been assigned B and constraint propagation has been done, as shown. MA has been chosen as the next variable to explore. List the values for MA that would be explored first by the Least-Constraining- Value Heuristic (LCV) _ **B**

AL	**BC**	**MA**	**NW**	**NU**	**ON**	**SA**	**YU**
B	RG	RGB	RG	RGB	RGB	RG	RGB