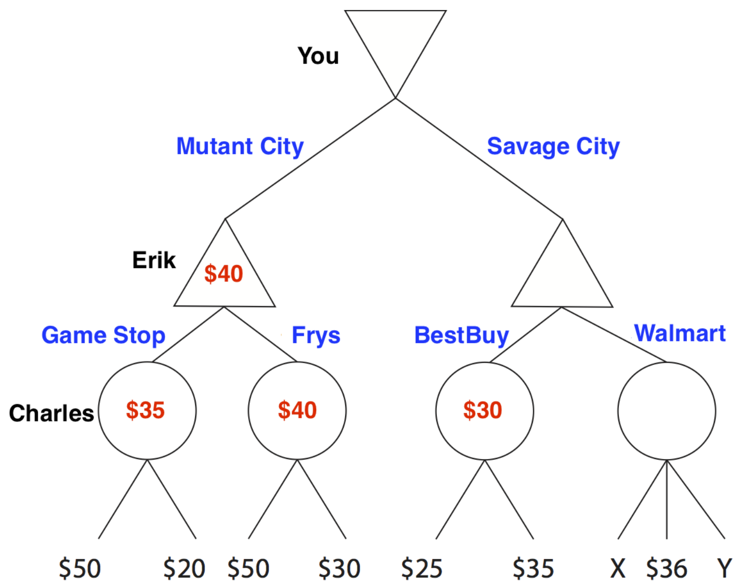
1. [15%] Birthday Gift

a) [5%]

Answer:
2% for the value of Professor T node.
1% for the value of each Charles node.

Each error -1%



b) [5%]

Answer:
1% for the value.
4% for the reason.
You will pick Mutant City if the value of the Savage City node is more than the value of the Mutant City node ($40), that is if:

$$\frac{x+y+36}{3} > 40 \Leftrightarrow x > 84 - y \Leftrightarrow x > 84$$

.

c) [5%]

Answer:
1% for the value.
4% for the reason.

BestBuy will be chosen if the value of the ”WalMart” node is less than the value of the ”BestBuy” node ($30):

$$\frac{x+y+36}{3} < 30 \Leftrightarrow x < 54 - y \Leftrightarrow x < 24$$

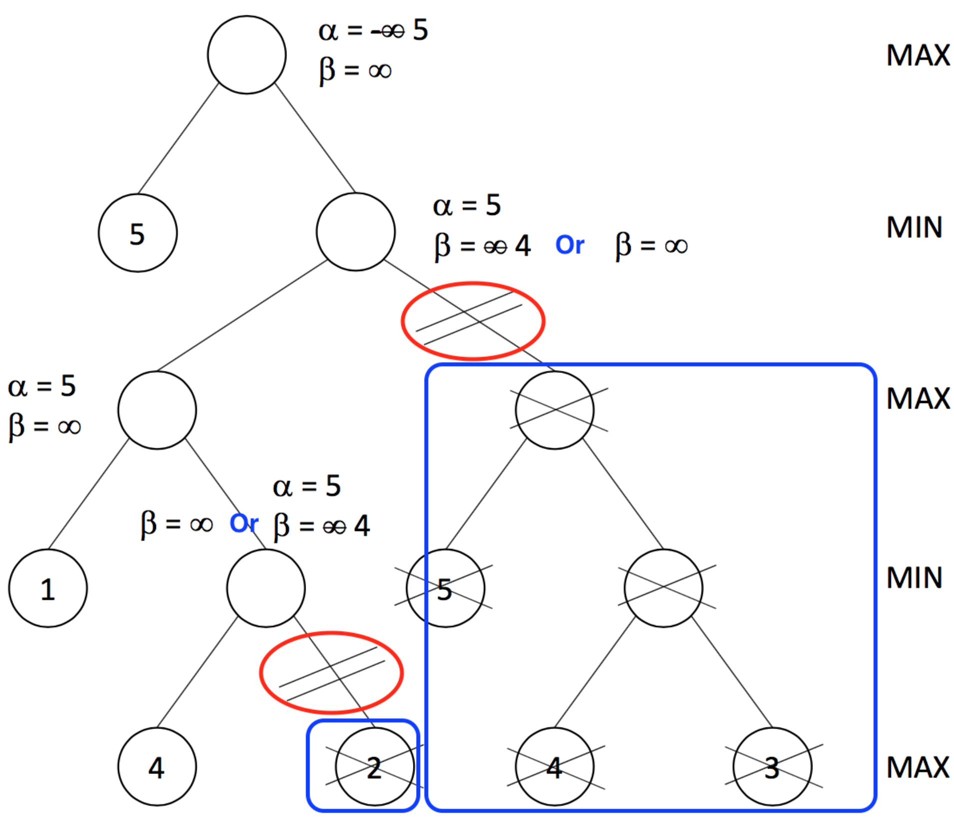
2. [20%] Game Playing

a1) [2%]

Answer:
The minimax value for the root node is 5.

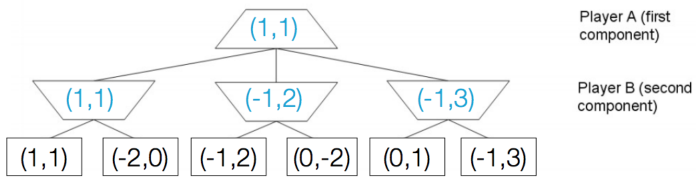
a2) [8%]

Answer:
1% for each "α β pair" of each node.
2% for each cut in the red circle.
Some students might cross out all the nodes under the cut in the red circle. Only if they cross out all the nodes in the corresponding blue rectangle, they get the point.



b1) [3%]

Answer:
1% for each pair in each node, no partial credit. In other words, if only 1 right (3 wrong), still 0 point.



b2) [4%]

Answer:
[4%] The values that the first and second player are trying to maximize are independent, so we no longer have situations where we know that one player will never let the other player down a particular branch of the game tree.
For instance, in the case where $U_A = U_B$, the problem reduces to searching for the max-valued leaf, which could appear anywhere in the tree.

b3) [3%]

Answer:
[1%] A's outcome can be worse than the computed v_A.

[2% for reasonable explain] For instance, in the example game, if B chooses (-2, 0) over (1, 1), then A's outcome will decrease from 1 to -2.

3. [30%] Search Algorithms

1) [5%]
Answer:
Nodes expanded: S, A **or** S, A, G1 **or** S, A, B, D, C, G1
Solution path: S, A, G1

2) [5%]
Answer:
Nodes Expanded: S, A, C **or** S, A, C, G1
Solution Path: S, A, C, G1

3) [5%]
Answer:
Nodes Expanded: S, A, B, D, E, C, H, F, G1
Solution Path: S, D, C, G1

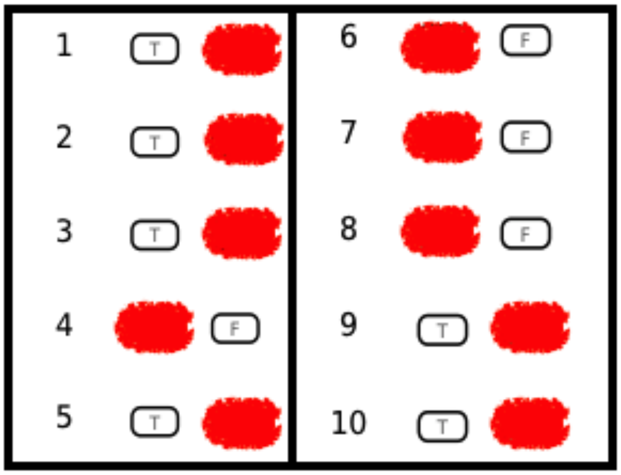
4) [5%]
Answer:
Nodes Expanded: S, B, E, H, G2
Solution Path: S, B, E, H, G2

5) [5%]
Answer:
Nodes Expanded: S, B, D, C, E, G1
Solution Path: S, D, C, G1

a) [3%]
Answer:
The heuristic is admissible. No node heuristic value overestimates the cost to reach the closest goal.

b) [2%]
Answer:
Yes, an admissible heuristic is needed to guarantee an optimal solution in A* tree search, while consistency is required when using graph search to guarantee optimality.

4. [10%] General AI Knowledge



5. [10%] Multiple Choice

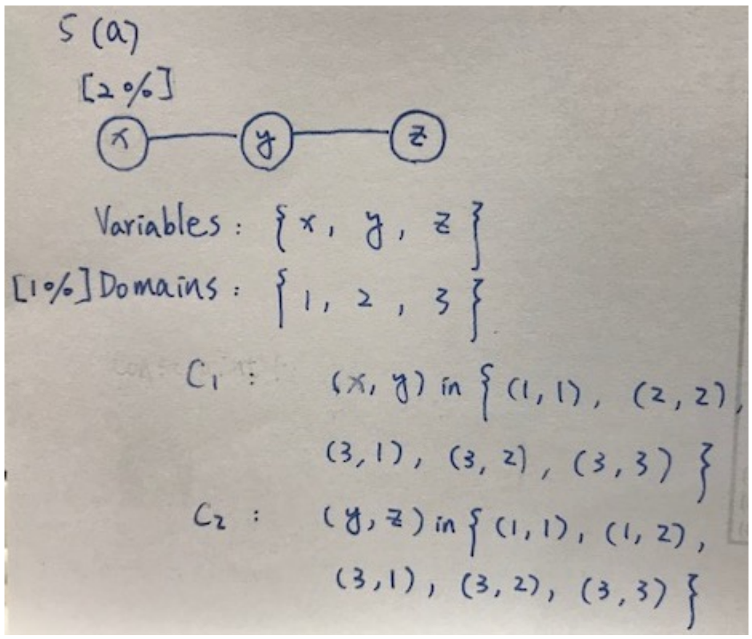
1	A [ ]	B [ ]	C [ ]	D [ ]	E [ ]
2	A [ ]	B [ ]	C [ ]	D [ ]	E [ ]
3	A [ ]	B [ ]	C [ ]	D [ ]	E [ ]
4	A [ ]	B [ ]	C [ ]	D [ ]	E [ ]
5	A [ ]	B [ ]	C [ ]	D [ ]	E [ ]

1. B
2. D
3. E
4. C
5. A

6. [15%] Constraint Satisfaction

a) [3%]

Answer:



b) [4%]

Answer:

[1%] Arc consistency first rules out $y = 2$,

[1%] because there is no value of z that satisfies C_2 if $y = 2$.

[1%] It then rules out $x = 2$,

[1%] because, once $y = 2$ is ruled out, there is no value of y that satisfies C_1 if $x = 2$.

Arc consistency does not rule out any other values at this point.

c) [4%]
Answer:
If we assign $x = 1$ during search, forward checking rules out [2%] $y = 3$ ($y = 2$ is already ruled out by arc consistency).

[2%] also rules out $z = 3$

d) [4%]

Answer:
Maintaining arc consistency rules out
[2%] $y = 3$
[2%] and $z = 3$.