1. [10%] General AI Knowledge

For each of the statements below, fill in the bubble T if the statement is always and unconditionally true, or fill in the bubble F if it is always false, sometimes false, or just does not make sense.

	T	F
1	T	●
2	T	●
3	●	F
~~4~~	T	●
5	●	F
~~6~~	●	F
7	●	F
8	T	●
9	T	●
10	●	F

1- Backgammon is deterministic.

2- In search, a state can have a parent.

3- An algorithm with a complexity of 3*n^2 and another one with a complexity of 5*n^2 + 3 have the same Big-O complexity.

4- In a problem formulation, "Start -> Goal" can be an operator.

5- If you know that the correct answer to this question is "T", then answering "T" to it is optimal.

6- A complete search algorithm will always find a solution. (if it exists!)

7- Even if all cars were self-driving, it would not make the world's roads a static environment.

8- The best chess programs in the world play chess optimally.

9- If all cars were self-driving, the world's roads would be a deterministic environment.

10- A first person shooter game (e.g. Halo, Quake, Call of Duty, Battlefield) played online between human players is not accessible to those human players.

Q2 Score: **30.0**

2. [30%] Search

Consider the search space on the following page, where S is the start state and ***D***, ***J***, and ***K*** satisfy the goal test. Arcs are labeled with the cost of traversing them and the estimated cost to a goal is reported inside nodes.

For each of the following search strategies, indicate which goal state is reached (if any) and list, in order, all the states of the nodes popped off of the OPEN queue, and the cost of the path found by the strategy to reach from S to the goal state. When all else is equal, nodes should be removed from OPEN in alphabetical order.

You should ***not expand nodes with states that have already been visited*** (where *"State X has been visited"* here means: some node was previously created and enqueued with state X). Note how the arcs in the figure are oriented, which means that you can only go from one state to another if the arrow points from the first to the second. For example, you can go from S to A (i.e., A is a successor of S) but not from A to S (i.e., S is not a successor of A).

a) [10%] Depth-first search

Goal state reached: J States popped off OPEN: SACHJ Path Cost 4 hops
(2+5+4+8 = 19)

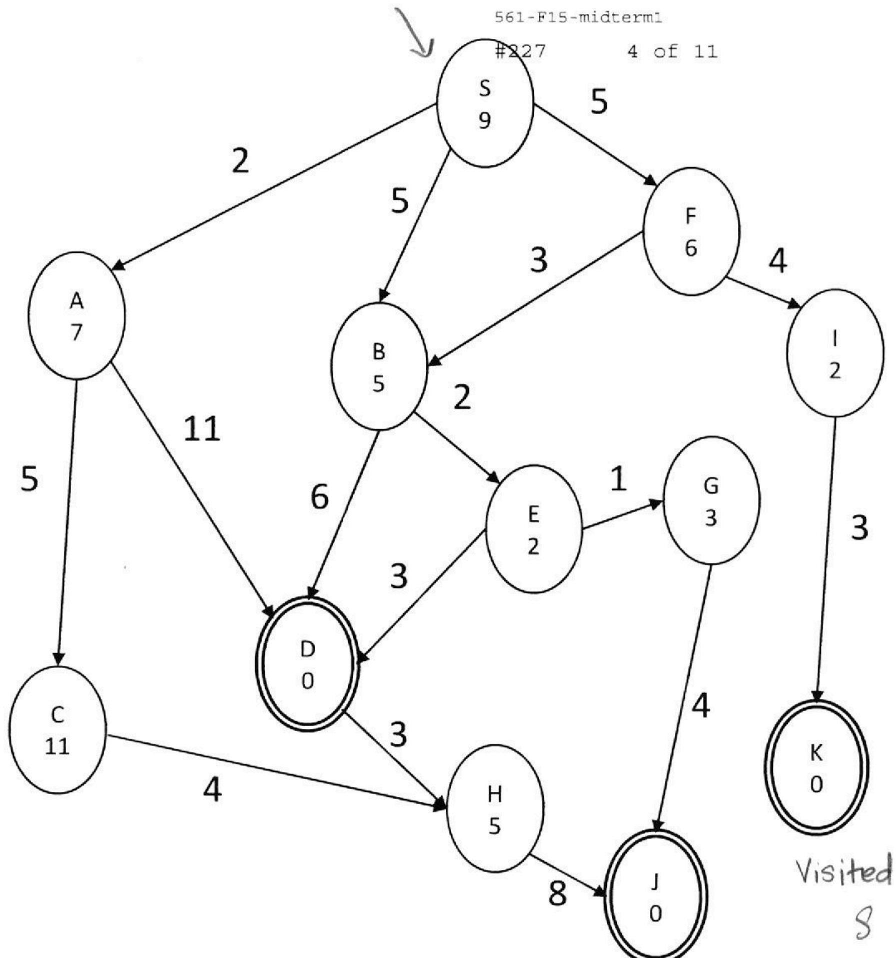
Cost = 19

b) [10%] Uniform cost Search

Goal state reached: D States popped off OPEN: SABFCEGID Path Cost 10
(S→B→E→D)

c) [10%] A* Search

Goal state reached: D States popped off OPEN: SABED Path Cost 10
(S→B→E→D)



DFS

Visited	Stack.
S	~~A~~ B F
A	~~C~~ D B F
C	~~H~~ D B F
H	~~J~~ D B F
J	

UCS

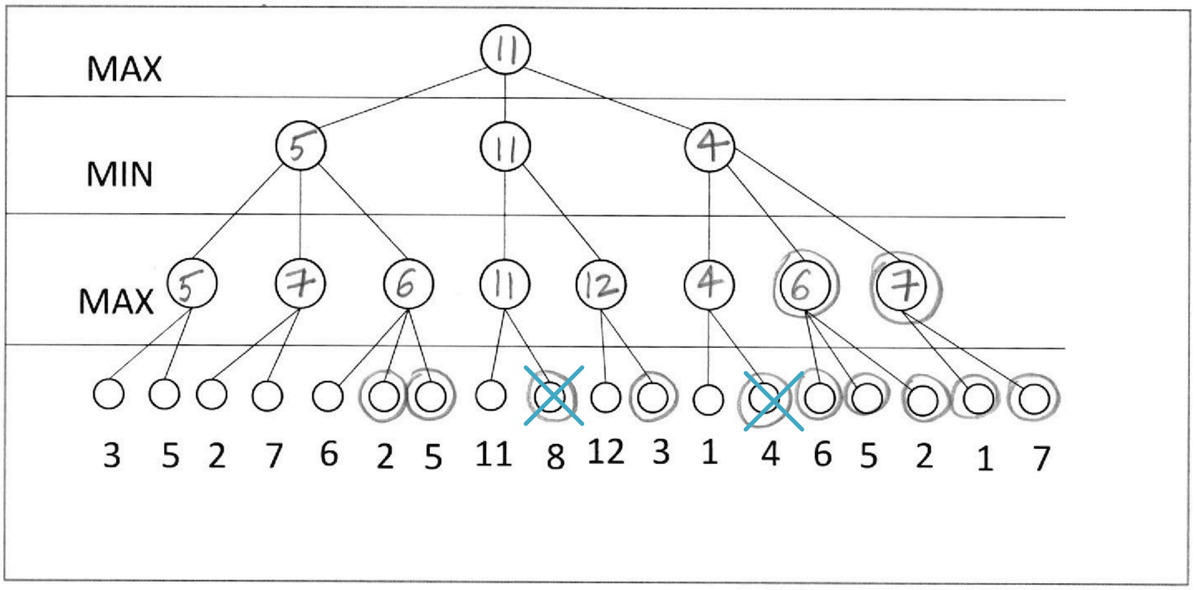
Visited	Queue
S	~~A(2)~~ B(5) F(5)
A	~~B(5)~~ F(5) C(7) D(13)
B	~~F(5)~~ C(7) E(7) D(11)
F	~~C(7)~~ E(7) I(9) D(11)
C	~~E(7)~~ I(9) D(11) H(11)
E	~~G(8)~~ I(9) D(10) H(11)
G	~~I(9)~~ D(10) H(11) J(12)
I	~~D(10)~~ H(11) J(12) K(12)
D	

A*

Visited	Queue.
S	~~A(9)~~ B(10) F(11)
A	~~B(10)~~ F(11) D(13) C(18)
B	~~E(9)~~ F(11) D(11) C(18)
E	~~D(10)~~ F(11) G(11) C(18)
D	

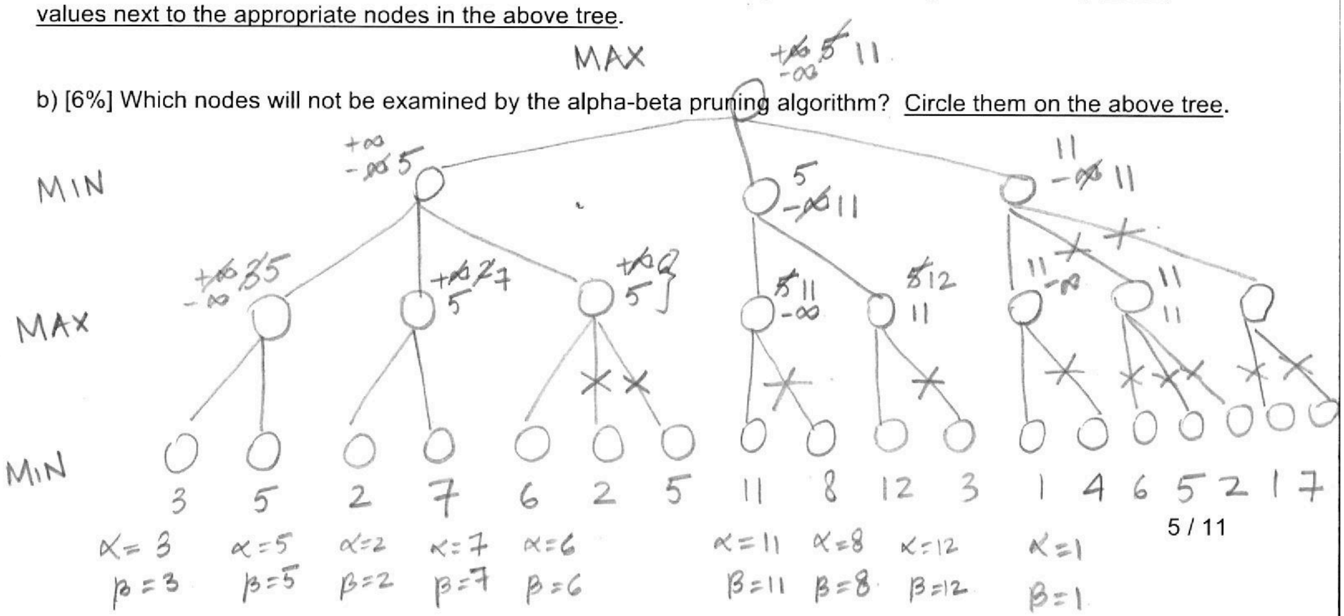
3. [10%] Game Playing

Consider the following game tree in which the evaluation function values are shown below each leaf node.
The root node is the maximizing player.
Assume that the search always visits children left-to-right.



a) [4%] Compute the backed-up values according to the minimax algorithm. Show your answer by writing values next to the appropriate nodes in the above tree.

b) [6%] Which nodes will not be examined by the alpha-beta pruning algorithm? Circle them on the above tree.



4. [20%] Constraint Satisfaction

You are in charge of a zoo.

Five animals need to be fed at specific times every **Monday, Wednesday and Friday**.

You need to assign **three zookeepers** on those days to feed the animals. The zookeepers have to wait until the feeding time is over. They have different skills, so some of them can only feed certain types of animals. They can also only feed one animal during a given time period.

Your animals need to be fed at the following times:

- Zebra: 6:00-6:30am
- Yabby: 6:30-7:00am
- X-Ray Fish: 6:45-7:15m
- Walrus: 6:15-6:45am
- Vervet: 6:15-6:30am

Your zookeepers have the following training:

- Ada can feed all animals except vervets.
- Bob can feed yabbies and walruses.
- Cecile can feed all animals.

a) Using the animal type as a variable, formulate this problem as a CSP with variables, domains and constraints. Constraints should be specified formally and precisely, but may be implicit rather than explicit.

Varibles → {Zebra, Yabby, X-Ray Fish, Walrus, Vervet}

Domains → {Ada, Bob, Cecile}

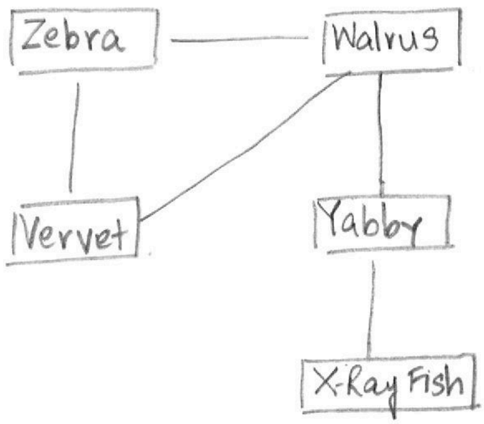
Constraints → 2 zookeepers cannot feed 2 animals at the same time & have constraints in training

Ada : {Zebra, Yabby, X-Ray Fish, Walrus}

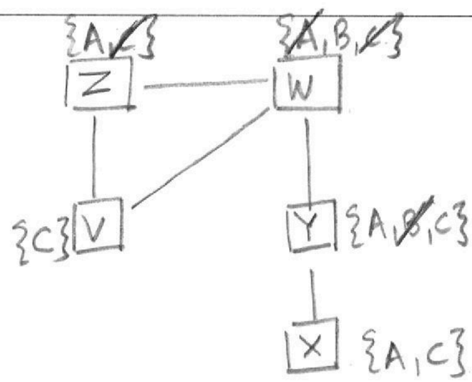
Bob : {Yabby, Walrus}

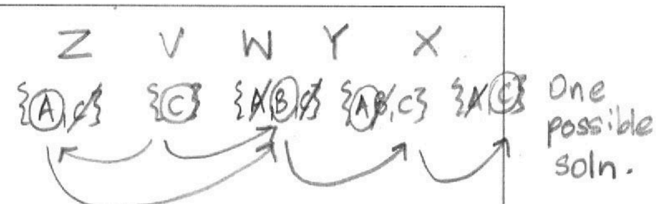
Cecile : {Zebra, Yabby, X-Ray Fish, Walrus, Vervet}

b) Draw the constraint graph.



c) Show the domains of the variables after running arc-consistency on this initial graph (after having already enforced any unary constraints).





After running arc consistency —

Zebra	Vervet	Walrus	Yabby
{Ada}	{Cecile}	{Bob}	{Ada}

X-Ray Fish {Cecile}

d) Give one solution to this CSP.

Zebra → Ada
Vervet → Cecile
Walrus → Bob
Yabby → Ada
X-Ray Fish → Cecile.

Q7 Score: **1.0**

5. [20%] Heuristic design for the Flip-Side game

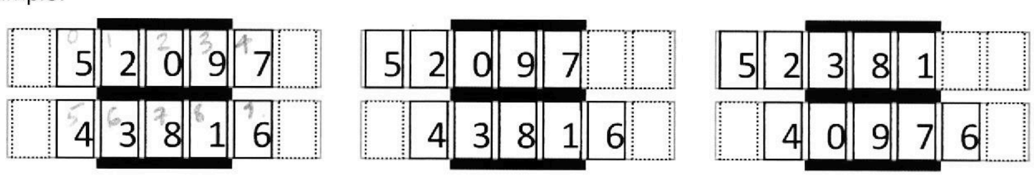
Flip-Side (source: http://jaapsch.net/puzzles/flipside.htm) is a game that consists of a frame containing 10 numbered blocks in ***two rows*** of five.

Each row contains additional space on either side of the blocks so that the blocks in the row can be slid either left or right. Note that all blocks slide together and no gaps can be formed within the blocks.

In the center of the frame is a flipper that holds 3 blocks from each row. When flipped, the three blocks in the top row swap with the three blocks directly beneath them.

The blocks are numbered from 0 to 9, and the aim is to arrange them in numerical order, i.e. 01234 on the top row, 56789 on the bottom row.

For example:



Initial Position Shift Top Left Flip

A. [2%] Describe a possible state representation for this game.

10 states → each corresponding to a numbered block from 0–9.

Therefore the state representation would be the position of each of the numbered blocks

states? needs clear explanation

B. [3%] Describe the operators to be used during the search for a solution to this game.

① Slide left

② Slide right

③ Flip (blocks in the center get swapped).

+1 for flip

C. [4%] What is the size of the state space, as a simple numerical formula?

10!

incomplete answer -2

Q9 Score: **6.0**

D. [1%] What kind of goal test will your search algorithm use?

all the blocks are in numerical order — 0, 1, 2, 3, 4 on top & 5, 6, 7, 8, 9 below

E. [10%] Provide two admissible heuristics (other than *h(n)=0* for all *n*) that can be used to guide an A* search in the Flip-Side game. Briefly explain why each heuristic is admissible [each heuristic formulation: 2%; each explanation: 3%].

① $\frac{10 - \#\text{ of ordered blocks}}{6}$

For the goal state, the # of ordered blocks would be = 10. Hence $h(n) = 0$ for the goal state. For every set of 3 #'s out of order, its taks a minimum of one move (using the flipper) to get them in order. Hence it is admissable.

\# of orderd blocks = 4

$\frac{10-4}{6} \geq 1$

0	6	7	8	4	
5	1	2	3	9	

↳ should be flipped = 1 move.

② $\frac{\#\text{ of unordered blocks}}{6}$

For the goal state, the # of unordered blocks = 0. Hence $h(n) = 0$ for goal state.

heuristics are not clear, neither the explanation

0 1 8 9 would need 2 left shifts & a flip,

5 6 2 3 4 = 3 moves

$h(n) = \frac{6}{6} = 1$, does not overestimate, hence is admissible.

6. [10%] AI Applications

Circle the **best** choice for each question:

(a) [2%] Eugene Goostman is famous for:
- a. Beating IBM Watson
- b. Passing the Turing Test
- c. Creating Google's self-driving car
- d. All of the above
- e. None of the above

(b) [2%] Genetic Algorithms are best suited to which problem domain?
- a. Travelling salesman problem
- b. N-queens problem
- c. Sudoku
- d. All of the above
- e. None of the above

regrade. +2

(c) [2%] Creating a rational AI agent guarantees that the agent will:
- a. Always win
- b. Be omniscient
- c. Optimize its performance measure
- d. All of the above
- e. None of the above

(d) [2%] For suboptimal route planning, Google Maps could use this search technique:
- a. Greedy Search
- b. A* with an inadmissible heuristic
- c. Depth-first search
- d. All of the above
- e. None of the above

(e) [2%] Assume that a rook can move on a chessboard any number of squares in a straight line, vertically or horizontally, but cannot jump over other pieces. Which heuristic is an admissible heuristic for the problem of moving the rook from square A to square B in the smallest number of moves?
- a. Straight-line distance
- b. Manhattan distance
- c. h(n)=1
- d. All of the above
- e. None of the above