1 Discussion Questions

(a) What is the difference between Forward Checking and AC-3?

Forward checking is like AC-3 but uses a more limited version of arc-consistency checking. Whenever a variable X is assigned, the forward-checking process establishes arc consistency for it. During forward checking, we are seeing if **a single arc is being consistent**, versus AC-3, which is **enforcing consistency for every arc**.

(b) Why does a tree-structured CSP take $O(nd^2)$ to solve?

If the constraint graph has no loops, the CSP can be solved in $O(nd^2)$ time compared to general CSPs, where worst-case time is $O(d^n)$. To solve a tree-structured CSP, first pick any variable to be the root of the tree, and choose an ordering of the variables such that each variable appears after its parent in the tree. Such an ordering is called a topological sort. Any tree with n nodes has $n-1$ arcs, so we can make this graph directed arc-consistent in $O(n)$ steps, each of which must compare up to d possible domain values for two variables, for a total time of $O(nd^2)$.

(c) Why would one use the following heuristics for CSP?

(i) Minimum Remaining Values (MRV)

MRV: Which variable next?

- Fail fast
- We have to assign all variables at some point, so we might as well do hard stuff first (which prunes a bunch of the search tree right away)

(ii) Least Constraining Value (LCV)

LCV: Which value next?

- We just want one solution.
- We dont try all combinations of value, so we should try ones that are likely to lead to a solution.

2 CSP: Air Traffic Control

We have five planes: A, B, C, D, and E and two runways: international and domestic. We would like to schedule a time slot and runway for each aircraft to either land or take off. We have four time slots: 1, 2, 3, 4 for each runway, during which we can schedule a landing or take off of a plane. We must find an assignment that meets the following constraints:

- Plane B has lost an engine and must land in time slot 1.
- Plane D can only arrive at the airport to land during or after time slot 3.
- Plane A is running low on fuel but can last until at most time slot 2.
- Plane D must land before plane C takes off, because some passengers must transfer from D to C.
- No two aircrafts can reserve the same time slot for the same runway.

(a) Complete the formulation of this problem as a CSP in terms of variables, domains, and constraints (both unary and binary). Constraints should be expressed implicitly using mathematical or logical notation rather than with words. Make sure to specify variables, domains, and constraints.

Variables: A, B, C, D, E for each plane.
Domains: a tuple (*runway type, time slot*) for runway type $\in$ {international, domestic} and time slot $\in \{1, 2, 3, 4\}$.
Constraints:

$$B[1] = 1$$

$$D[1] \geq 3$$

$$A[1] \leq 2$$

$$D[1] < C[1]$$

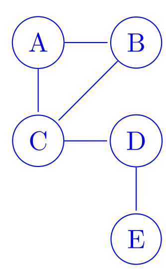
$$A \neq B \neq C \neq D \neq E$$

For the following parts, we add the following two constraints:

- Planes A, B, and C cater to international flights and can only use the international runway.
- Planes D and E cater to domestic flights and can only use the domestic runway.

(b) With the addition of the two constraints above, we completely reformulate the CSP. You are given the variables and domains of the new formulation. Complete the constraint graph for this problem given the original constraints and the two added ones

Variables: A, B, C, D, E for each plane.
Domains: $\{1, 2, 3, 4\}$
Constraints Graph:



Explanation of Constraints Graph: We can now encode the runway information into the identity of the variable, since each runway has more than enough time slots for the planes it serves. We represent the non-colliding time slot constraint as a binary constraint between the planes that use the same runways.

(c) What are the domains of the variables after enforcing arc-consistency? Begin by enforcing unary constraints. (Cross out values that are no longer in the domain.)

A	~~1~~	2	~~3~~	~~4~~
B	1	~~2~~	~~3~~	~~4~~
C	~~1~~	~~2~~	~~3~~	4
D	~~1~~	~~2~~	3	~~4~~
E	1	2	~~3~~	4

(d) Arc-consistency can be rather expensive to enforce, and we believe that we can obtain faster solutions using only forward-checking on our variable assignments. Using the Minimum Remaining Values heuristic, perform backtracking search on the graph, breaking ties by picking lower values and characters first. List the (variable, assignment) pairs in the order they occur (including the assignments that are reverted upon reaching a dead end). Enforce unary constraints before starting the search.

List of (variable, assignment) pairs:

(You don't have to use this table)

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

Answer: (B, 1), (A, 2), (C, 3), (C, 4), (D, 3), (E, 1)

3 Cargo Plane: Linear Programming Formulation

A cargo plane has three compartments for storing cargo: front, centre and rear. These compartments have the following limits on both weight and space:

Compartment	Weight capacity (tonnes)	Space capacity (cubic metres)
Front	10	6800
Centre	16	8700
Rear	8	5300

The following four cargoes are available for shipment on the next flight:

Cargo	Weight (tonnes)	Volume (cubic metres/tonne)	Profit ($/tonne)
C1	18	480	310
C2	15	650	380
C3	23	580	350
C4	12	390	285

Any proportion of these cargoes can be accepted. The objective is to determine how much of each cargo C1, C2, C3 and C4 should be accepted and how to distribute each among the compartments so that the total profit for the flight is maximised. **Formulate** the above problem as a linear program. Think about the assumptions you are making when formulating this problem as a linear program.

Variables:

We need to decide how much of each of the four cargoes to put in each of the three compartments. Hence let $x_{i,j}$ be the number of tonnes of cargo i (i=1,2,3,4 for C1, C2, C3 and C4 respectively) that is put into compartment j (j=1 for Front, j=2 for Centre and j=3 for Rear) where $x_{i,j} \geq 0; i = 1, 2, 3, 4; j = 1, 2, 3$.

(Note here that we are explicitly told we can split the cargoes into any proportions (fractions) that we like.)

Constraints:

1. We cannot pack more of each of the four cargoes than we have available.

$$x_{1,1} + x_{1,2} + x_{1,3} \leq 18$$

$$x_{2,1} + x_{2,2} + x_{2,3} \leq 15$$

$$x_{3,1} + x_{3,2} + x_{3,3} \leq 23$$

$$x_{4,1} + x_{4,2} + x_{4,3} \leq 12$$

2. The weight capacity of each compartment must be respected.

$$x_{1,1} + x_{2,1} + x_{3,1} + x_{4,1} \leq 10$$

$$x_{1,2} + x_{2,2} + x_{3,2} + x_{4,2} \leq 16$$

$$x_{1,3} + x_{2,3} + x_{3,3} + x_{4,3} \leq 8$$

3. The volume (space) capacity of each compartment must be respected.

$$480x_{1,1} + 650x_{2,1} + 580x_{3,1} + 390x_{4,1} \leq 6800$$

$$480x_{1,2} + 650x_{2,2} + 580x_{3,2} + 390x_{4,2} \leq 8700$$

$$480x_{1,3} + 650x_{2,3} + 580x_{3,3} + 390x_{4,3} \leq 5300$$

Objective: The objective is to maximise total profit, i.e.
maximise $310(x_{1,1} + x_{1,2} + x_{1,3}) + 380(x_{2,1} + x_{2,2} + x_{2,3}) + 350(x_{3,1} + x_{3,2} + x_{3,3}) + 285(x_{4,1} + x_{4,2} + x_{4,3})$
The basic assumptions are:

1. that each cargo can be split into whatever proportions/fractions we desire

2. that each cargo can be split between two or more compartments if we so desire
3. that the cargo can be packed into each compartment (for example if the cargo was spherical it would not be possible to pack a compartment to volume capacity, some free space is inevitable in sphere packing)
4. all the data/numbers given are accurate