Name	Prithivi Raj	Matric number	A0173060R
Collaborator		Collaborator's matric number	
Tutorial			

Instructions:

- *Your solutions for this tutorial must be TYPE-WRITTEN.*
- *Print and submit your solution(s) to your tutor. You can make another copy for yourself if necessary. Late submission will NOT be entertained.*
- *YOUR SOLUTION TO QUESTION* 3 *will be GRADED for this tutorial.*
- *You can work in pairs, but each of you should submit the solution(s) individually.*
- *Include the name of your collaborator in your submission.*

1. Consider the *item allocation problem*. We have a group of people $N = \{1, \dots, n\}$, and a group of items $G = \{g_1, \dots, g_m\}$. Each person $i \in N$ has a utility function $u_i : G \to \mathbb{R}_+$. The constraint is that every person is assigned *at most one item*, and each item is assigned to *at most one person*. An allocation simply says which person gets which item (if any). In what follows, you *must* use *only* the binary variables $x_{i,j} \in \{0, 1\}$, where $x_{i,j} = 1$ if person i receives the good g_j, and is 0 afterwards.

   (a) Write out the constraints: 'each person receives no more than one item' and 'each item goes to at most one person', using only the $x_{i,j}$ variables[1].

   (b) Suppose that people are divided into *disjoint types* $N_1, \dots, N_k$ (think of, say, genders or ethnicities), and items are divided into *disjoint blocks* $G_1, \dots, G_\ell$. We require that every person type N_p is allowed to take no more than λ_{pq} items from block G_q. Write out this constraint using the $x_{i,j}$ variables.

   (c) We say that player i *envies* player i' if the utility that player i has from his assigned item is strictly lower than the utility that player i might have received from the item assigned to player i'. Write out the constraints that ensure that in the allocation, no player envies any other player. You may assume that the validity constraints from (a) hold.

   Solution:

   (a) <u>Constraints</u>

   - $x_{i,j} \in \{0, 1\}$
   - If $x_{i,j} > 0$ then, $x_{i,j} > x_{i,k}$ for any $g_k \in G$
   - If $x_{i,j} > 0$ then, $x_{i,j} > x_{z,j}$ for any $z \in N$

[1] You may use simple algebraic functions $-, +, \times, \div$, and numbers

(b) Added Constraint

- For $a \in N_x$ where $x \in \{1, ..., m\}$ and $b \in G_y$ where $y \in \{1, ..., q\}$

$$\lambda_{ab} \geq x_{a,b} \geq 0$$

(c) Added Constraint

- For $a \in N_x$ where $x \in \{1, ..., m\}$ and $b \in G_y$ where $y \in \{1, ..., q\}$

$$\lambda_{ab} \geq x_{a,b} \geq 0$$

2. A discrete and finite domain constraint satisfaction problem consists of a finite set of *variables*, $\mathcal{V} = \{V_1, ..., V_n\}$; for each variable $X \in V$ a finite domain of *values*, $Dom(X) = \{x_1, ..., x_k\}$; and a finite collection of *constraints*, $\mathcal{C} = \{C_1, \ldots, C_m\}$. Each constraint $C \in \mathcal{C}$ is a constraint over some set of variables $Vars(C)$. The size of this set is known as the *arity* of the constraint. Given the CSP defined below, answer the following questions:

   - variables: $(A,\ Dom(A) = \{1, 2, 3\})$, $(B,\ Dom(B) = \{2, 3, 6\})$, $(C,\ Dom(C) = \{2, 3, 5, 6\})$
   - constraints $C_1 : A + B = C,\ C_2 : A \leq B,\ C_3 : B \leq C.$

   (a) Depict the CSP as a graph (you are free to choose the vertex and edge set as per your choice).

   (b) Depict the CSP as a hyper-graph.

   (c) Given any CSP with V variables and C constraints, all of which are n-ary ($n \geq 3$), what is the minimum number of vertices required to represent it in a graph?

   Solution:

   Your solution here

3. Consider a CSP with variables $x_1, \ldots, x_6$ where $dom(x_i) = \{1, 2, 3, 4, 5, 6\} \quad \forall i \in [1, 6]$. The CSP contains the following constraints:

   - $C_1 : x_4 \geq x_1 + 3$
   - $C_2 : x_4 \geq x_2 + 3$
   - $C_3 : x_5 \geq x_3 + 3$
   - $C_4 : x_5 \geq x_4 + 1$
   - $C_5 :$ *allDiff* $\{x_1, x_2, x_3, x_4, x_5\}$

   Mention the *bound-consistent* domain for each variable in this CSP.

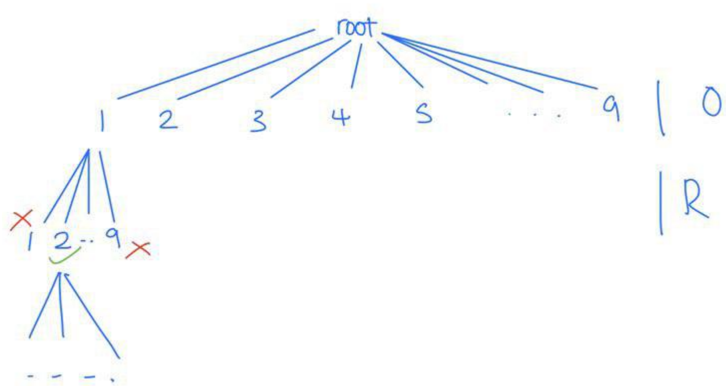
   Solution:

- $dom(x_1) = \{1, 2, 3\}$
- $dom(x_2) = \{1, 2, 3\}$
- $dom(x_3) = \{1, 2, 3\}$
- $dom(x_4) = \{5, 6\}$
- $dom(x_5) = \{4, 5, 6\}$
- $dom(x_6) = \{1, 2, 3, 4, 5, 6\}$

4. Show a trace of the backtracking algorithm with forward checking to solve the cryptarithmetic problem shown in Figure 1. Use the most constrained variable heuristic, and assume that the domain values (digits) are tried in ascending order (i.e., 0, 1, 2, $\cdots$).

Figure 1: Cryptarithmetic puzzle.

Solution:



Backtracking does not spawn the sub-tree of nodes that do not adhere to the given constraints.