# Assignment 2

## Task 1

|  |
| --- |
| **Part a**  Constraint graph for the Map problem, |
| **Part b**  To implement backtracking search and to use Degree heuristic, determining the number of constraints for each of these states.   |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | State | Ri | Re | S | C | W | D | N | V | I | | Number of Constraints | 5 | 5 | 3 | 3 | 2 | 2 | 2 | 2 | 0 |   Implementing MRV and Degree heuristic to determine the constraint satisfaction,   |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **State** | **Ri** | **Re** | **S** | **C** | **W** | **D** | **N** | **V** | **I** | | Initial | R G B | R G B | R G B | R G B | R G B | R G B | R G B | R G B | R G B | | From the Initial state, choosing the variable based on the Degree heuristic.  Degree heuristic : Re, Ri  Choosing **Re**  Setting Re = B | | | | | | | | | | | **State** | **Ri** | **Re** | **S** | **C** | **W** | **D** | **N** | **V** | **I** | | Level 1 | R G | B | R G | R G | R G | R G | R G B | R G B | R G B | | Variables with Minimum Remaining value **:** Ri, S, C, W, D  Based on Degree heuristic for MRV variables : **Ri**  Setting Ri = R | | | | | | | | | | | **State** | **Ri** | **Re** | **S** | **C** | **W** | **D** | **N** | **V** | **I** | | Level 2 | R | B | R G | G | G | R G | G B | G B | R G B | | Variables with Minimum Remaining value **:** C, W  Based on Degree heuristic for MRV variables : **C**  C is a single valued variable. Hence not making a new assignment.   |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **State** | **Ri** | **Re** | **S** | **C** | **W** | **D** | **N** | **V** | **I** | | Level 3 | R | B | R G | G | G | R G | G B | G B | R G B |   Variables with Minimum Remaining value **:** **W**  W is a single valued variable. Hence not making a new assignment.   |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **State** | **Ri** | **Re** | **S** | **C** | **W** | **D** | **N** | **V** | **I** | | Level 4 | R | B | R G | G | G | R G | G B | G B | R G B |   Variables with Minimum Remaining value **:** S, D, N, V  Based on Degree heuristic for MRV variables : **S**  Setting S = R (Setting S = G will not satisfy the condition C = G , hence setting the other option S= R ) | | | | | | | | | | | **State** | **Ri** | **Re** | **S** | **C** | **W** | **D** | **N** | **V** | **I** | | Level 5 | R | B | R | G | G | G | G B | G B | R G B | | Variables with Minimum Remaining value **:** **D**  D is a single valued variable. Hence not making a new assignment.   |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **State** | **Ri** | **Re** | **S** | **C** | **W** | **D** | **N** | **V** | **I** | | Level 6 | R | B | R | G | G | G | G B | G B | R G B |   Variables with Minimum Remaining value **:** N, V  Degree heuristic for MRV variables : **N**  Setting N = B | | | | | | | | | | | **State** | **Ri** | **Re** | **S** | **C** | **W** | **D** | **N** | **V** | **I** | | Level 7 | R | B | R | G | G | G | B | G | R G B | | Variables with Minimum Remaining value **:** **V**  V is a single valued variable. Hence not making a new assignment.   |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **State** | **Ri** | **Re** | **S** | **C** | **W** | **D** | **N** | **V** | **I** | | Level 8 | R | B | R | G | G | G | B | G | R G B |   Variables with Minimum Remaining Value : **I**  Choosing I **=** R | | | | | | | | | | | **State** | **Ri** | **Re** | **S** | **C** | **W** | **D** | **N** | **V** | **I** | | Level 9 | R | B | R | G | G | G | B | G | R |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **State** | **Ri** | **Re** | **S** | **C** | **W** | **D** | **N** | **V** | **I** | | Solution | R | B | R | G | G | G | B | G | R |   The above solution is achieved by applying MRV and degree heuristic at each level of the search. If there had been any failures during this assignment, the constraint assignment would be reverted to the level at which the assignment has been made and a different assignment would have been made that would satisfy the constraint conditions. |
| **Part c**   |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **State** | **Ri** | **Re** | **S** | **C** | **W** | **D** | **N** | **V** | **I** | | Initial | R | R G B | R G B | R G B | R G B | R G B | R G B | R G B | R G B |   As per the problem statement, Red is assigned to the first variable Ri in Part b. Based on the assignment of Red to Ri, the following constraint modification happens,  Ri is the first variable in Part b, ( Re with 5 constraints), assigning Ri = R   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | When Ri = R, then arcs to be verified are as follows,  N -> Ri  V -> Ri  C ->Ri  Re -> Ri  W -> Ri   |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **Ri** | **Re** | **S** | **C** | **W** | **D** | **N** | **V** | **I** | | R | R GB | R GB | R GB | R GB | R GB | R GB | R GB | R GB | | N -> Ri  V -> Ri  C ->Ri  Re -> Ri  W -> Ri | | **Verifying N->Ri**  When N is B and G, then Ri has a valid value R  When N is R, then Ri does not have a valid value. So, N cannot be R  Removing R from N.  So adding the following arcs for validations  V->N  Ri->N   |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **Ri** | **Re** | **S** | **C** | **W** | **D** | **N** | **V** | **I** | | R | R GB | R GB | R GB | R GB | R GB | GB | R GB | R GB |   Verification completed for N->Ri | ~~N -> Ri~~  V -> Ri  C ->Ri  Re -> Ri  W -> Ri  V->N  Ri->N | | **Verifying V->Ri**  When V is B and G, then Ri has a valid value R  When V is R, then Ri does not have a valid value. So, V cannot be R  Removing R from V.  So adding the following arcs for validations  N->V  Ri->V   |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **Ri** | **Re** | **S** | **C** | **W** | **D** | **N** | **V** | **I** | | R | R GB | R GB | R GB | R GB | R GB | GB | GB | R GB |   Verification completed for V->Ri | ~~N -> Ri~~  ~~V -> Ri~~  C ->Ri  Re -> Ri  W -> Ri  V->N  Ri->N  N->V  Ri->V | | **Verifying C-> Ri**  When C is B and G, then Ri has a valid value R  When C is R, then Ri does not have a valid value. So, C cannot be R  Removing R from C.  So adding the following arcs for validations  Ri ->C  Re-> C  S->C   |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **Ri** | **Re** | **S** | **C** | **W** | **D** | **N** | **V** | **I** | | R | R GB | R GB | GB | R GB | R GB | GB | GB | R GB |   Verification completed for C-> Ri | ~~N -> Ri~~  ~~V -> Ri~~  ~~C ->Ri~~  Re -> Ri  W -> Ri  V->N  Ri->N  N->V  Ri->V  Ri ->C  Re-> C  S->C | | **Verifying Re->Ri**  When Re is B and G, then Ri has a valid value R  When Re is R, then Ri does not have a valid value. So, Re cannot be R  Removing R from Re.  So adding the following arcs for validations  W->Re  S->Re  Ri->Re  C->Re  D->Re   |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **Ri** | **Re** | **S** | **C** | **W** | **D** | **N** | **V** | **I** | | R | GB | R GB | GB | R GB | R GB | GB | GB | R GB |   Verification completed for Re-> Ri | ~~N -> Ri~~  ~~V -> Ri~~  ~~C ->Ri~~  ~~Re -> Ri~~  W -> Ri  V->N  Ri->N  N->V  Ri->V  Ri ->C  Re-> C  S->C  W->Re  S->Re  Ri->Re  C->Re  D->Re | | **Verifying W->Ri**  When W is B and G, then Ri has a valid value R  When W is R, then Ri does not have a valid value. So, W cannot be R  Removing R from W.  So adding the following arcs for validations  Ri->W  Re->W   |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **Ri** | **Re** | **S** | **C** | **W** | **D** | **N** | **V** | **I** | | R | GB | R GB | GB | GB | R GB | GB | GB | R GB |   Verification completed for W-> Ri | ~~N -> Ri~~  ~~V -> Ri~~  ~~C ->Ri~~  ~~Re -> Ri~~  ~~W -> Ri~~  V->N  Ri->N  N->V  Ri->V  Ri ->C  Re-> C  S->C  W->Re  S->Re  Ri->Re  C->Re  D->Re  Ri->W  Re->W | | **Verifying V->N**  When V is B, then N has a valid value G  When V is G, then N has a valid value B  This satisfies the arc consistency validation.  No new arcs are added.  No modification in the constraints.   |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **Ri** | **Re** | **S** | **C** | **W** | **D** | **N** | **V** | **I** | | R | GB | R GB | GB | GB | R GB | GB | GB | R GB |   Verification completed for V->N | ~~N -> Ri~~  ~~V -> Ri~~  ~~C ->Ri~~  ~~Re -> Ri~~  ~~W -> Ri~~  ~~V->N~~  Ri->N  N->V  Ri->V  Ri ->C  Re-> C  S->C  W->Re  S->Re  Ri->Re  C->Re  D->Re  Ri->W  Re->W | | **Verifying Ri->N**  When Ri is R , then N has a valid value GB  This satisfies the arc consistency validation.  No new arcs are added.  No modification in the constraints.   |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **Ri** | **Re** | **S** | **C** | **W** | **D** | **N** | **V** | **I** | | R | GB | R GB | GB | GB | R GB | GB | GB | R GB |   Verification completed for Ri->N | ~~N -> Ri~~  ~~V -> Ri~~  ~~C ->Ri~~  ~~Re -> Ri~~  ~~W -> Ri~~  ~~V->N~~  ~~Ri->N~~  N->V  Ri->V  Ri ->C  Re-> C  S->C  W->Re  S->Re  Ri->Re  C->Re  D->Re  Ri->W  Re->W | | **Verifying N->V**  When N is B, then V has a valid value G  When N is G, then V has a valid value B  This satisfies the arc consistency validation.  No new arcs are added.  No modification in the constraints.   |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **Ri** | **Re** | **S** | **C** | **W** | **D** | **N** | **V** | **I** | | R | GB | R GB | GB | GB | R GB | GB | GB | R GB |   Verification completed for N->V | ~~N -> Ri~~  ~~V -> Ri~~  ~~C ->Ri~~  ~~Re -> Ri~~  ~~W -> Ri~~  ~~V->N~~  ~~Ri->N~~  ~~N->V~~  Ri->V  Ri ->C  Re-> C  S->C  W->Re  S->Re  Ri->Re  C->Re  D->Re  Ri->W  Re->W | | **Verifying Ri->V**  When Ri is R , then V has a valid value GB  This satisfies the arc consistency validation.  No new arcs are added.  No modification in the constraints.   |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **Ri** | **Re** | **S** | **C** | **W** | **D** | **N** | **V** | **I** | | R | GB | R GB | GB | GB | R GB | GB | GB | R GB |   Verification completed for Ri->V | ~~N -> Ri~~  ~~V -> Ri~~  ~~C ->Ri~~  ~~Re -> Ri~~  ~~W -> Ri~~  ~~V->N~~  ~~Ri->N~~  ~~N->V~~  ~~Ri->V~~  Ri ->C  Re-> C  S->C  W->Re  S->Re  Ri->Re  C->Re  D->Re  Ri->W  Re->W | | **Verifying Ri->C**  When Ri is R , then C has valid value GB  This satisfies the arc consistency validation.  No new arcs are added.  No modification in the constraints.   |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **Ri** | **Re** | **S** | **C** | **W** | **D** | **N** | **V** | **I** | | R | GB | R GB | GB | GB | R GB | GB | GB | R GB |   Verification completed for Ri->C | ~~N -> Ri~~  ~~V -> Ri~~  ~~C ->Ri~~  ~~Re -> Ri~~  ~~W -> Ri~~  ~~V->N~~  ~~Ri->N~~  ~~N->V~~  ~~Ri->V~~  ~~Ri ->C~~  Re-> C  S->C  W->Re  S->Re  Ri->Re  C->Re  D->Re  Ri->W  Re->W | | **Verifying Re->C**  When Re is B, then C has a valid value G  When Re is G, then C has a valid value B  This satisfies the arc consistency validation.  No new arcs are added.  No modification in the constraints.   |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **Ri** | **Re** | **S** | **C** | **W** | **D** | **N** | **V** | **I** | | R | GB | R GB | GB | GB | R GB | GB | GB | R GB |   Verification completed for Re->C | ~~N -> Ri~~  ~~V -> Ri~~  ~~C ->Ri~~  ~~Re -> Ri~~  ~~W -> Ri~~  ~~V->N~~  ~~Ri->N~~  ~~N->V~~  ~~Ri->V~~  ~~Ri ->C~~  ~~Re-> C~~  S->C  W->Re  S->Re  Ri->Re  C->Re  D->Re  Ri->W  Re->W | | **Verifying S->C**  When S is B, then C has a valid value G  When S is G, then C has a valid value B  When S is R, then C has a valid value B  This satisfies the arc consistency validation.  No new arcs are added.  No modification in the constraints.   |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **Ri** | **Re** | **S** | **C** | **W** | **D** | **N** | **V** | **I** | | R | GB | R GB | GB | GB | R GB | GB | GB | R GB |   Verification completed for S->C | ~~N -> Ri~~  ~~V -> Ri~~  ~~C ->Ri~~  ~~Re -> Ri~~  ~~W -> Ri~~  ~~V->N~~  ~~Ri->N~~  ~~N->V~~  ~~Ri->V~~  ~~Ri ->C~~  ~~Re-> C~~  ~~S->C~~  W->Re  S->Re  Ri->Re  C->Re  D->Re  Ri->W  Re->W | | **Verifying W->Re**  When W is B, then Re has a valid value G  When W is G, then Re has a valid value B  When W is R, then Re has a valid value B  This satisfies the arc consistency validation.  No new arcs are added.  No modification in the constraints.   |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **Ri** | **Re** | **S** | **C** | **W** | **D** | **N** | **V** | **I** | | R | GB | R GB | GB | GB | R GB | GB | GB | R GB |   Verification completed for W->Re | ~~N -> Ri~~  ~~V -> Ri~~  ~~C ->Ri~~  ~~Re -> Ri~~  ~~W -> Ri~~  ~~V->N~~  ~~Ri->N~~  ~~N->V~~  ~~Ri->V~~  ~~Ri ->C~~  ~~Re-> C~~  ~~S->C~~  ~~W->Re~~  S->Re  Ri->Re  C->Re  D->Re  Ri->W  Re->W | | **Verifying S->Re**  When S is B, then Re has a valid value G  When S is G, then Re has a valid value B  When S is R, then Re has a valid value B  This satisfies the arc consistency validation.  No new arcs are added.  No modification in the constraints.   |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **Ri** | **Re** | **S** | **C** | **W** | **D** | **N** | **V** | **I** | | R | GB | R GB | GB | GB | R GB | GB | GB | R GB |   Verification completed for S->Re | ~~N -> Ri~~  ~~V -> Ri~~  ~~C ->Ri~~  ~~Re -> Ri~~  ~~W -> Ri~~  ~~V->N~~  ~~Ri->N~~  ~~N->V~~  ~~Ri->V~~  ~~Ri ->C~~  ~~Re-> C~~  ~~S->C~~  ~~W->Re~~  ~~S->Re~~  Ri->Re  C->Re  D->Re  Ri->W  Re->W | | **Verifying Ri->Re**  When Ri is R, then Re has a valid value B  This satisfies the arc consistency validation.  No new arcs are added.  No modification in the constraints.   |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **Ri** | **Re** | **S** | **C** | **W** | **D** | **N** | **V** | **I** | | R | GB | R GB | GB | GB | R GB | GB | GB | R GB |   Verification completed for Ri->Re | ~~N -> Ri~~  ~~V -> Ri~~  ~~C ->Ri~~  ~~Re -> Ri~~  ~~W -> Ri~~  ~~V->N~~  ~~Ri->N~~  ~~N->V~~  ~~Ri->V~~  ~~Ri ->C~~  ~~Re-> C~~  ~~S->C~~  ~~W->Re~~  ~~S->Re~~  ~~Ri->Re~~  C->Re  D->Re  Ri->W  Re->W | | **Verifying C->Re**  When C is B, then Re has a valid value G  When C is G, then Re has a valid value B  This satisfies the arc consistency validation.  No new arcs are added.  No modification in the constraints.   |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **Ri** | **Re** | **S** | **C** | **W** | **D** | **N** | **V** | **I** | | R | GB | R GB | GB | GB | R GB | GB | GB | R GB |   Verification completed for C->Re | ~~N -> Ri~~  ~~V -> Ri~~  ~~C ->Ri~~  ~~Re -> Ri~~  ~~W -> Ri~~  ~~V->N~~  ~~Ri->N~~  ~~N->V~~  ~~Ri->V~~  ~~Ri ->C~~  ~~Re-> C~~  ~~S->C~~  ~~W->Re~~  ~~S->Re~~  ~~Ri->Re~~  ~~C->Re~~  D->Re  Ri->W  Re->W | | **Verifying D->Re**  When D is B, then Re has a valid value G  When D is G, then Re has a valid value B  When D is R, then Re has a valid value B  This satisfies the arc consistency validation.  No new arcs are added.  No modification in the constraints.   |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **Ri** | **Re** | **S** | **C** | **W** | **D** | **N** | **V** | **I** | | R | GB | R GB | GB | GB | R GB | GB | GB | R GB |   Verification completed for D->Re | ~~N -> Ri~~  ~~V -> Ri~~  ~~C ->Ri~~  ~~Re -> Ri~~  ~~W -> Ri~~  ~~V->N~~  ~~Ri->N~~  ~~N->V~~  ~~Ri->V~~  ~~Ri ->C~~  ~~Re-> C~~  ~~S->C~~  ~~W->Re~~  ~~S->Re~~  ~~Ri->Re~~  ~~C->Re~~  ~~D->Re~~  Ri->W  Re->W | | **Verifying Ri->W**  When Ri is R, then W has a valid value B  This satisfies the arc consistency validation.  No new arcs are added.  No modification in the constraints.   |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **Ri** | **Re** | **S** | **C** | **W** | **D** | **N** | **V** | **I** | | R | GB | R GB | GB | GB | R GB | GB | GB | R GB |   Verification completed for Ri->W | ~~N -> Ri~~  ~~V -> Ri~~  ~~C ->Ri~~  ~~Re -> Ri~~  ~~W -> Ri~~  ~~V->N~~  ~~Ri->N~~  ~~N->V~~  ~~Ri->V~~  ~~Ri ->C~~  ~~Re-> C~~  ~~S->C~~  ~~W->Re~~  ~~S->Re~~  ~~Ri->Re~~  ~~C->Re~~  ~~D->Re~~  ~~Ri->W~~  Re->W | | **Verifying Re->W**  When Re is B, then W has a valid value G  When Re is G, then W has a valid value B  This satisfies the arc consistency validation.  No new arcs are added.  No modification in the constraints.   |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **Ri** | **Re** | **S** | **C** | **W** | **D** | **N** | **V** | **I** | | R | GB | R GB | GB | GB | R GB | GB | GB | R GB |   Verification completed for Re->W | ~~N -> Ri~~  ~~V -> Ri~~  ~~C ->Ri~~  ~~Re -> Ri~~  ~~W -> Ri~~  ~~V->N~~  ~~Ri->N~~  ~~N->V~~  ~~Ri->V~~  ~~Ri ->C~~  ~~Re-> C~~  ~~S->C~~  ~~W->Re~~  ~~S->Re~~  ~~Ri->Re~~  ~~C->Re~~  ~~D->Re~~  ~~Ri->W~~  ~~Re->W~~ |   The arc consistency check for Ri= R has completed successfully.  Inference: For the constraint Ri= R, all other states can have a valid value without any failures. At the end of this arc consistency check, below is the state of the variables.   |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **States** | **Ri** | **Re** | **S** | **C** | **W** | **D** | **N** | **V** | **I** | | **Stage 2** | R | G B | R G B | G B | G B | R G B | G B | G B | R G B |   Choosing the next variable based on degree heuristic ( Re with 5 constraints), assigning Re = G   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | When Re = G, then arcs to be verified are as follows,  W -> Re  Ri -> Re  C ->Re  S -> Re  D -> Re   |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **Ri** | **Re** | **S** | **C** | **W** | **D** | **N** | **V** | **I** | | R | G | R GB | G B | G B | R GB | G B | G B | R GB | | W -> Re  Ri -> Re  C ->Re  S -> Re  D -> Re | | **Verifying W-> Re**  When W is B, then Re has a valid value G  When W is G, then Re does not have a valid value. So, W cannot be G  Removing G from W.  So adding the following arcs for validations  Ri ->W  Re-> W   |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **Ri** | **Re** | **S** | **C** | **W** | **D** | **N** | **V** | **I** | | R | G | R GB | G B | B | R GB | G B | G B | R GB |   Verification completed for W-> Re | ~~W -> Re~~  Ri -> Re  C ->Re  S -> Re  D -> Re  Ri ->W  Re-> W | | **Verifying Ri -> Re**  When Ri is R, then Re has a valid value G  This satisfies the arc consistency validation.  No new arcs are added.  No modification in the constraints.   |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **Ri** | **Re** | **S** | **C** | **W** | **D** | **N** | **V** | **I** | | R | G | R GB | G B | B | R GB | G B | G B | R GB |   Verification completed for Ri -> Re | ~~W -> Re~~  ~~Ri -> Re~~  C ->Re  S -> Re  D -> Re  Ri ->W  Re-> W | | **Verifying C-> Re**  When C is B, then Re has a valid value G  When C is G, then Re does not have a valid value. So, C cannot be G  Removing G from C.  So adding the following arcs for validations  Ri ->C  Re-> C  S->C   |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **Ri** | **Re** | **S** | **C** | **W** | **D** | **N** | **V** | **I** | | R | G | R GB | B | G | R GB | G B | G B | R GB |   Verification completed for C-> Re | ~~W -> Re~~  ~~Ri -> Re~~  ~~C ->Re~~  S -> Re  D -> Re  Ri ->W  Re-> W  Ri ->C  Re->C  S->C | | **Verifying S-> Re**  When S is B, then Re has a valid value G  When S is R, then Re has a valid value G  When S is G, then Re does not have a valid value. So, S cannot be G  Removing G from S.  So adding the following arcs for validations  C-> S  Re->S  D ->S   |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **Ri** | **Re** | **S** | **C** | **W** | **D** | **N** | **V** | **I** | | R | G | R B | B | G | R GB | G B | G B | R GB |   Verification completed for S-> Re | ~~W -> Re~~  ~~Ri -> Re~~  ~~C ->Re~~  ~~S -> Re~~  D -> Re  Ri ->W  Re-> W  Ri ->C  Re-> C  S->C  C-> S  Re->S  D ->S | | **Verifying D-> Re**  When D is B, then Re has a valid value G  When D is R, then Re has a valid value G  When D is G, then Re does not have a valid value. So, D cannot be G  Removing G from D.  So adding the following arcs for validations  S->D  Re->D   |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **Ri** | **Re** | **S** | **C** | **W** | **D** | **N** | **V** | **I** | | R | G | R B | B | G | R B | G B | G B | R GB |   Verification completed for D-> Re | ~~W -> Re~~  ~~Ri -> Re~~  ~~C ->Re~~  ~~S -> Re~~  ~~D -> Re~~  Ri ->W  Re-> W  Ri ->C  Re-> C  S->C  C-> S  Re->S  D ->S  S->D  Re->D | | **Verifying Ri->W**  When Ri is R, then W has a valid value B  This satisfies the arc consistency validation.  No new arcs are added.  No modification in the constraints.   |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **Ri** | **Re** | **S** | **C** | **W** | **D** | **N** | **V** | **I** | | R | G | R B | B | B | R B | G B | G B | R GB |   Verification completed for Ri->W | ~~W -> Re~~  ~~Ri -> Re~~  ~~C ->Re~~  ~~S -> Re~~  ~~D -> Re~~  ~~Ri ->W~~  Re-> W  Ri ->C  Re-> C  S->C  C-> S  Re->S  D ->S  S->D  Re->D | | **Verifying Re->W**  When Re is G, then W has a valid value B  This satisfies the arc consistency validation.  No new arcs are added.  No modification in the constraints.   |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **Ri** | **Re** | **S** | **C** | **W** | **D** | **N** | **V** | **I** | | R | G | R B | B | B | R B | G B | G B | R GB |   Verification completed for Re->W | ~~W -> Re~~  ~~Ri -> Re~~  ~~C ->Re~~  ~~S -> Re~~  ~~D -> Re~~  ~~Ri ->W~~  ~~Re-> W~~  Ri ->C  Re-> C  S->C  C-> S  Re->S  D ->S  S->D  Re->D | | **Verifying Ri->C**  When Ri is R , then C has a valid value B  This satisfies the arc consistency validation.  No new arcs are added.  No modification in the constraints.   |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **Ri** | **Re** | **S** | **C** | **W** | **D** | **N** | **V** | **I** | | R | G | R B | B | B | R B | G B | G B | R GB |   Verification completed for Ri->C | ~~W -> Re~~  ~~Ri -> Re~~  ~~C ->Re~~  ~~S -> Re~~  ~~D -> Re~~  ~~Ri ->W~~  ~~Re-> W~~  ~~Ri ->C~~  Re-> C  S->C  C-> S  Re->S  D ->S  S->D  Re->D | | **Verifying Re->C**  When Re is G, then C has a valid value B  This satisfies the arc consistency validation.  No new arcs are added.  No modification in the constraints.   |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **Ri** | **Re** | **S** | **C** | **W** | **D** | **N** | **V** | **I** | | R | G | R B | B | B | R B | G B | G B | R GB |   Verification completed for Re->C | ~~W -> Re~~  ~~Ri -> Re~~  ~~C ->Re~~  ~~S -> Re~~  ~~D -> Re~~  ~~Ri ->W~~  ~~Re-> W~~  ~~Ri ->C~~  ~~Re-> C~~  S->C  C-> S  Re->S  D ->S  S->D  Re->D | | **Verifying S->C**  When S is R, then C has a valid value B  When S is B, then C does not have a valid value. So, S cannot be B  Removing B from S.  So adding the following arcs for validations  Re->S  D->S   |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **Ri** | **Re** | **S** | **C** | **W** | **D** | **N** | **V** | **I** | | R | G | R | B | B | R B | G B | G B | R GB |   Verification completed for S->C | ~~W -> Re~~  ~~Ri -> Re~~  ~~C ->Re~~  ~~S -> Re~~  ~~D -> Re~~  ~~Ri ->W~~  ~~Re-> W~~  ~~Ri ->C~~  ~~Re-> C~~  ~~S->C~~  C-> S  Re->S  D ->S  S->D  Re->D  Re->S  D->S | | **Verifying C->S**  When C is B, then S has a valid value R  This satisfies the arc consistency validation.  No new arcs are added.  No modification in the constraints.   |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **Ri** | **Re** | **S** | **C** | **W** | **D** | **N** | **V** | **I** | | R | G | R | B | B | R B | G B | G B | R GB |   Verification completed for C->S | ~~W -> Re~~  ~~Ri -> Re~~  ~~C ->Re~~  ~~S -> Re~~  ~~D -> Re~~  ~~Ri ->W~~  ~~Re-> W~~  ~~Ri ->C~~  ~~Re-> C~~  ~~S->C~~  ~~C-> S~~  Re->S  D ->S  S->D  Re->D  Re->S  D->S | | **Verifying Re->S**  When Re is G, then S has a valid value R  This satisfies the arc consistency validation.  No new arcs are added.  No modification in the constraints.   |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **Ri** | **Re** | **S** | **C** | **W** | **D** | **N** | **V** | **I** | | R | G | R | B | B | R B | G B | G B | R GB |   Verification completed for Re->S | ~~W -> Re~~  ~~Ri -> Re~~  ~~C ->Re~~  ~~S -> Re~~  ~~D -> Re~~  ~~Ri ->W~~  ~~Re-> W~~  ~~Ri ->C~~  ~~Re-> C~~  ~~S->C~~  ~~C-> S~~  ~~Re->S~~  D ->S  S->D  Re->D  Re->S  D->S | | **Verifying D->S**  When D is B, then S has a valid value R  When D is R, then S has no valid value. So D cannot be R  Removing R from D.  So adding the following arcs for validations  Re->D  S->D  S->D and Re->D are duplicates that are already available for validation in the arc. So not adding duplicates to the arc.   |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **Ri** | **Re** | **S** | **C** | **W** | **D** | **N** | **V** | **I** | | R | G | R | B | B | B | G B | G B | R GB |   Verification completed for D->S | ~~W -> Re~~  ~~Ri -> Re~~  ~~C ->Re~~  ~~S -> Re~~  ~~D -> Re~~  ~~Ri ->W~~  ~~Re-> W~~  ~~Ri ->C~~  ~~Re-> C~~  ~~S->C~~  ~~C-> S~~  ~~Re->S~~  ~~D ->S~~  S->D  Re->D  Re->S  D->S | | **Verifying S->D**  When S is R, then D has a valid value B  When S is B, then D has a valid value R  This satisfies the arc consistency validation.  No new arcs are added.  No modification in the constraints.   |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **Ri** | **Re** | **S** | **C** | **W** | **D** | **N** | **V** | **I** | | R | G | R | B | B | B | G B | G B | R GB |   Verification completed for S->D | ~~W -> Re~~  ~~Ri -> Re~~  ~~C ->Re~~  ~~S -> Re~~  ~~D -> Re~~  ~~Ri ->W~~  ~~Re-> W~~  ~~Ri ->C~~  ~~Re-> C~~  ~~S->C~~  ~~C-> S~~  ~~Re->S~~  ~~D ->S~~  ~~S->D~~  Re->D  Re->S  D->S | | **Verifying Re->D**  When Re is G, then D has a valid value R B  This satisfies the arc consistency validation.  No new arcs are added.  No modification in the constraints.   |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **Ri** | **Re** | **S** | **C** | **W** | **D** | **N** | **V** | **I** | | R | G | R | B | B | B | G B | G B | R GB |   Verification completed for Re->D | ~~W -> Re~~  ~~Ri -> Re~~  ~~C ->Re~~  ~~S -> Re~~  ~~D -> Re~~  ~~Ri ->W~~  ~~Re-> W~~  ~~Ri ->C~~  ~~Re-> C~~  ~~S->C~~  ~~C-> S~~  ~~Re->S~~  ~~D ->S~~  ~~S->D~~  ~~Re->D~~  Re->S  D->S | | **Verifying Re->S**  When Re is G, then S has a valid value R B  This satisfies the arc consistency validation.  No new arcs are added.  No modification in the constraints.   |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **Ri** | **Re** | **S** | **C** | **W** | **D** | **N** | **V** | **I** | | R | G | R | B | B | B | G B | G B | R GB |   Verification completed for Re->S | ~~W -> Re~~  ~~Ri -> Re~~  ~~C ->Re~~  ~~S -> Re~~  ~~D -> Re~~  ~~Ri ->W~~  ~~Re-> W~~  ~~Ri ->C~~  ~~Re-> C~~  ~~S->C~~  ~~C-> S~~  ~~Re->S~~  ~~D ->S~~  ~~S->D~~  ~~Re->D~~  ~~Re->S~~  D->S | | **Verifying D->S**  When Re is G, then S has a valid value R B  This satisfies the arc consistency validation.  No new arcs are added.  No modification in the constraints.   |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **Ri** | **Re** | **S** | **C** | **W** | **D** | **N** | **V** | **I** | | R | G | R | B | B | B | G B | G B | R GB |   Verification completed for Re->S | ~~W -> Re~~  ~~Ri -> Re~~  ~~C ->Re~~  ~~S -> Re~~  ~~D -> Re~~  ~~Ri ->W~~  ~~Re-> W~~  ~~Ri ->C~~  ~~Re-> C~~  ~~S->C~~  ~~C-> S~~  ~~Re->S~~  ~~D ->S~~  ~~S->D~~  ~~Re->D~~  ~~Re->S~~  ~~D->S~~ |   The arc consistency check for Re= G has completed successfully.  Inference: For the constraint Re= G , all other states can have a valid value without any failures. At the end of this arc consistency check, below is the state of the variables.   |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **Ri** | **Re** | **S** | **C** | **W** | **D** | **N** | **V** | **I** | | R | G | R | B | B | B | G B | G B | R GB |   Based on degree heuristic assigning the next constraint to V (number of constraints : 2)   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | When V = B, then arcs to be verified are as follows,  N->V  Ri->V   |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **Ri** | **Re** | **S** | **C** | **W** | **D** | **N** | **V** | **I** | | R | G | R GB | G B | G B | R GB | G B | B | R GB | | N->V  Ri->V | | **Verifying N->V**  When N is G , then V has a valid value B  When N is B, then V does not have a valid value. So, N cannot be B  Removing B from N.  So adding the following arcs for validations  Ri->N  V->N   |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **Ri** | **Re** | **S** | **C** | **W** | **D** | **N** | **V** | **I** | | R | G | R | B | B | B | G | B | R GB |   Verification completed for N->V | ~~N->V~~  Ri->V  Ri->N  V->N | | **Verifying Ri->V**  When Ri is R , then V has a valid value B  This satisfies the arc consistency validation.  No new arcs are added.  No modification in the constraints.   |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **Ri** | **Re** | **S** | **C** | **W** | **D** | **N** | **V** | **I** | | R | G | R | B | B | B | G | B | R GB |   Verification completed for Ri->V | ~~N->V~~  ~~Ri->V~~  Ri->N  V->N | | **Verifying Ri->N**  When Ri is R , then N has a valid value G  This satisfies the arc consistency validation.  No new arcs are added.  No modification in the constraints.   |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **Ri** | **Re** | **S** | **C** | **W** | **D** | **N** | **V** | **I** | | R | G | R | B | B | B | G | B | R GB |   Verification completed for Ri->N | ~~N->V~~  ~~Ri->V~~  ~~Ri->N~~  V->N | | **Verifying V->N**  When V is B, then N has a valid value G  This satisfies the arc consistency validation.  No new arcs are added.  No modification in the constraints.   |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **Ri** | **Re** | **S** | **C** | **W** | **D** | **N** | **V** | **I** | | R | G | R | B | B | B | G | B | R GB |   Verification completed for V->N | ~~N->V~~  ~~Ri->V~~  ~~Ri->N~~  ~~V->N~~ |   The arc consistency check for V = B has completed successfully.  Inference: For the constraint V = B, all other states can have a valid value without any failures. At the end of this arc consistency check, below is the state of the variables.   |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **Ri** | **Re** | **S** | **C** | **W** | **D** | **N** | **V** | **I** | | R | G | R | B | B | B | G | B | R GB |   I is an independent state with no arc associated to it. So assigning constraints to I will not violate the constraint satisfaction condition, hence satisfying the constraints.  Hence assigning I = G   |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **Ri** | **Re** | **S** | **C** | **W** | **D** | **N** | **V** | **I** | | R | G | R | B | B | B | G | B | G |   Thus the constraints are satisfied using arc consistency. |
| **Part d**  **Implementing Problem structure**  **1.Independent subproblems**  From the constraint graph developed in Part a, we can see that I is a separate subproblem whose constraint is not affected by any other variables in the problem.  So the two sets of the sub problem are as follows,   |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **Ri** | **Re** | **S** | **C** | **W** | **D** | **N** | **V** | **I** |   The nodes coloured in grey and violet form two sub problems based on the concept of Independent subproblems.  **2. Implementing Cutset Conditioning**  To convert the sub problem comprising of Ri, Re, S, C, W, D, N and V as a linear tree structure the following nodes marked in grey are to be removed based on the concept of cutset conditioning.  Removing the nodes N and Re will convert the constraint graph into a tree structured CSP without any loops.  Considering Ri as the root of the tree, below is the resulting tree structure,  Let us assign colours Red to N and Re. That leaves the tree structure with the below options for assignment of constraint.  B G  B G  B G  B G  B G  B G    Assigning the colour Blue to the root node Ri,  Ri = Blue  B G  B G  B G  B G  B G  That leaves the option Green for V, W and C,  V = Green, W = Green and C = Green  B G  B G  This leaves S with the option Blue. S = Blue  B G  Since S is blue, D will be Green. D = Green  Thus satisfying all the constraint for Sub problem 1. Assigning a constraint for Sub problem 2 also since it has no dependency on other variables.  We can see that with implementation of Problem structure, constraint satisfaction is achieved more efficiently with less number of variables to handle. It simplifies the number of Iterations needed to verify if the conditions for constraints are satisfied. |
| **Part e**  Valid solution for this map based on the Arc consistency explained in Part c |

## Task 2

|  |
| --- |
| For this cryptarithmetic problem, below are the attributes for computing the constraint graph,  Variables: **S E N D M O R Y X1 X2 X3 X4**  Domains : {0,1,2,3,4,5,6,7,8,9}  Constraints :  D + E = Y + (10 \* X1)  N + R + X1 = E + (10 \* X2)  E + O + X2 = N + (10 \* X3)  S + M + X3 = O + (10 \* X4)  X4 = M  **Constraint Graph** |