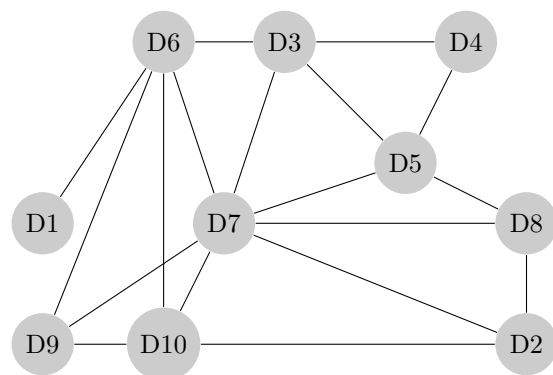


Problem 1. A graph of the CSP



Variables: Domains

D1: District 1 {blue, chartreuse, green, red}
D2 : District2 {blue, chartreuse, green, red}
D3 : District3 {blue, chartreuse, green, red}
D4 : District4 {blue, chartreuse, green, red}
D5 : District5 {blue, chartreuse, green, red}
D6 : District6 {blue, chartreuse, green, red}
D7 : District7 {blue, chartreuse, green, red}
D8 : District8 {blue, chartreuse, green, red}
D9 : District9 {blue, chartreuse, green, red}
D10 : District10 {blue, chartreuse, green, red}

Constraints

$D1 \neq D6$, $D2 \neq D7$, $D2 \neq D8$, $D2 \neq D10$, $D3 \neq D6$, $D3 \neq D7$, $D3 \neq D4$,
 $D3 \neq D5$, $D4 \neq D5$, $D5 \neq D8$, $D5 \neq D7$, $D6 \neq D7$, $D6 \neq D9$, $D6 \neq D10$,
 $D7 \neq D9$, $D7 \neq D8$, $D7 \neq D10$, $D9 \neq D10$

Problem 2

D7 would be chosen since it has 6 constraints which is the most in the constraint problem $D7 \neq D2$, $D7 \neq D3$, $D7 \neq D5$, $D7 \neq D6$, $D7 \neq D8$, $D7 \neq D9$

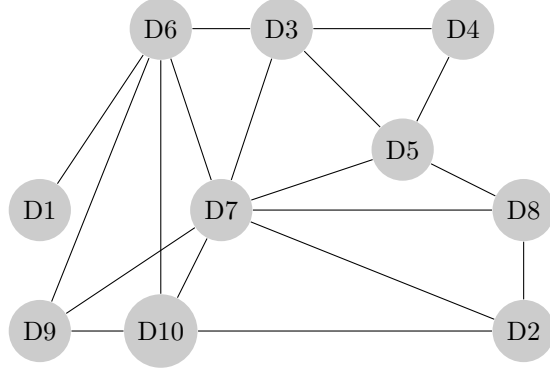
Problem 3 Variables: Domains

D1: District 1 {blue, chartreuse, green, red}
D2 : District2 {blue, chartreuse, green, red}
D3 : District3 {blue, chartreuse, green, red}
D4 : District4 {blue, chartreuse, green, red}
D5 : District5 {blue, chartreuse, green, red}
D6 : District6 {blue, chartreuse, green, red}
D7 : District7 {blue}

D8 : District8 {blue, chartreuse, green, red}
D9 : District9 {blue, chartreuse, green, red}
D10 : District10 {blue, chartreuse, green, red}

Revise(csp, District 2, District 7)
 District 2 = {blue, chartreuse, green, red} District7 = {blue}
x = blue
CheckConstraint if x ≠ District7(false)
remove x from District2 domain
 District2 = {chartreuse, green, red}
x = chartreuse
CheckConstraint if x ≠ District7(true)
Keep x in Domain
 District2 = {chartreuse, green, red}
x = green
CheckConstraint if x ≠ District7(true)
Keep x in Domain
 District2 = {chartreuse, green, red}
x = red CheckConstraint if x ≠ District7(true)
Keep x in Domain

Problem 4 District 2 is the next node to consider
 It is assigned chartreuse because removing any other color will make both
 neighboring domains 2, so alphabetical choice is made.



Domains

D1: District 1 {blue, chartreuse, green, red}
D2 : District2 {chartreuse}
D3 : District3 {chartreuse, green, red}
D4 : District4 {blue, chartreuse, green, red}
D5 : District5 {chartreuse, green, red}
D6 : District6 {chartreuse, green, red}

D7 : District7 {blue}
D8 : District8 {chartreuse, green, red}
D9 : District9 {chartreuse, green, red}
D10 : District10 {chartreuse, green, red}

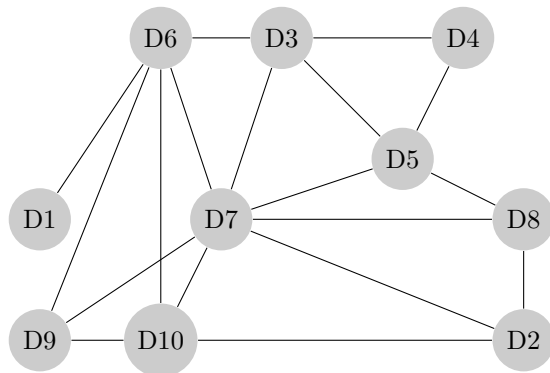
NowAfterArcconsistency

D1 : District1 {blue, chartreuse, green, red}
D2 : District2 {chartreuse}
D3 : District3 {chartreuse, green, red}
D4 : District4 {blue, chartreuse, green, red}
D5 : District5 {chartreuse, green, red}
D6 : District6 {chartreuse, green, red}
D7 : District7 {blue}
D8 : District8 {green, red}
D9 : District9 {chartreuse, green, red}
D10 : District10 {green, red}

chartreuse is removed as domain value from neighbors of D2

Problem 5

1. D8 is next pick green
Thus D2 domain is $\{chartreuse, red\}$
2. D5 is selected as chartreuse
D3 domain is $\{green, red\}$
3. D3 is selected as green
D4 domain is $\{blue, red\}$
4. D4 is selected as blue
5. D6 is selected as chartreuse
D1 domain is $\{blue, red\}$
D9domainis {green, red}
6. D9 is selected as green
D10 domain is $\{red\}$
7. D10 is selected as red
D1 domain is $\{blue\}$
8. D1 is set to blue Final Graph and Domains is



D1: District 1 {*blue*}
 D2 : District2 {*chartreuse*}
 D3 : District3 {*green*}
 D4 : District4 {*blue*}
 D5 : District5 {*chartreuse*}
 D6 : District6 {*chartreuse*}
 D7 : District7 {*blue*}
 D8 : District8 {*green*}
 D9 : District9 {*green*}
 D10 : District10 {*red*}

If we didn't use forward checking we would end up in a fail state more often because we chose a new state without considering the future possible conflicts (consequences)