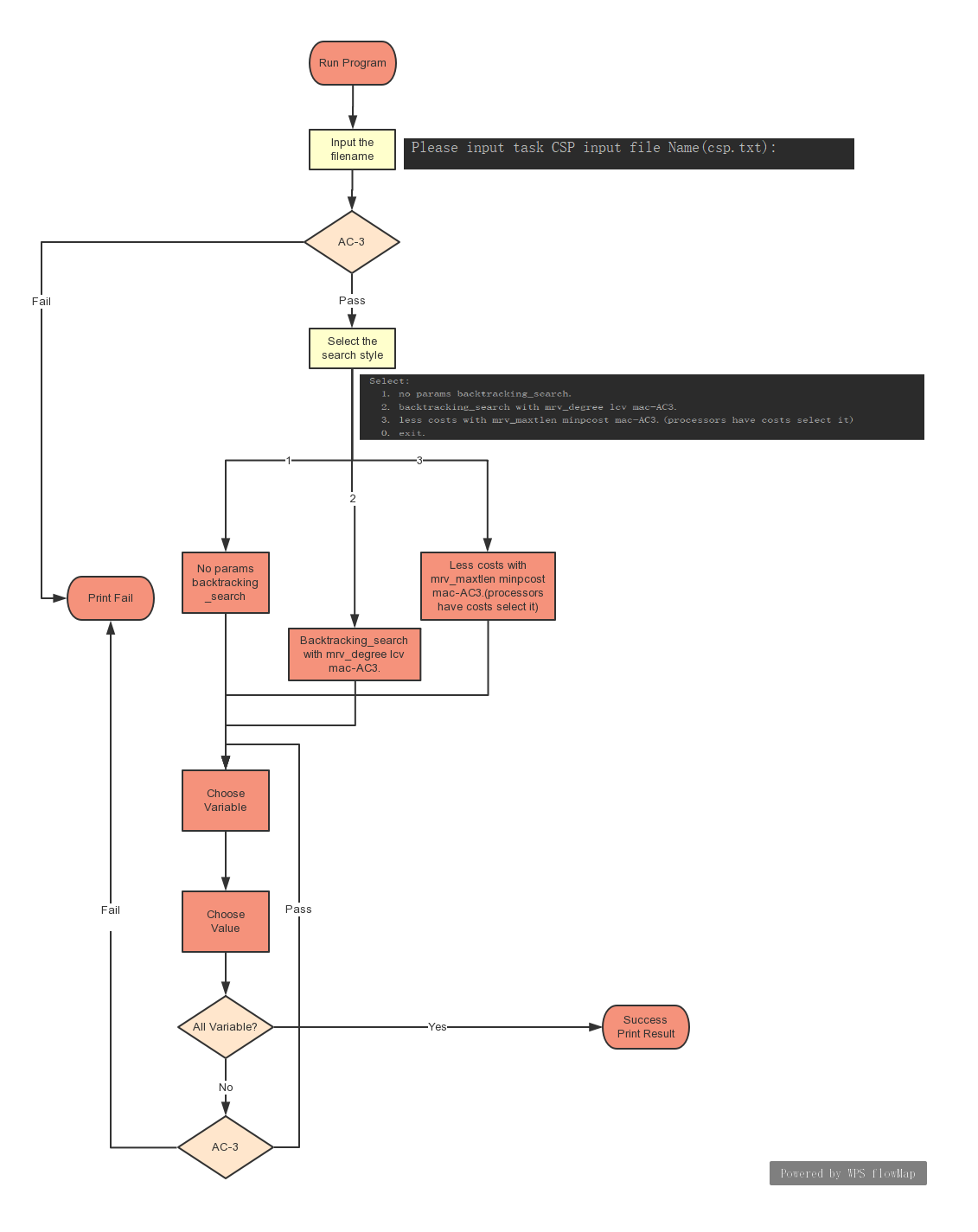
**Project Part 1: CSP solver**

Jiayi Li, Yichen Li

1. **Instructions**
   1. Put the test file and the python file in the same directory.
   2. Run the python file (cspsolver.py).
   3. Input test file name according to the instruction. For example, if the name of the test file is test1, then input “test1.txt”.
   4. After reading the file, input “1”,”2” or “3” according to the instruction. If you input “1”, the backtrack approach will be executed without any heuristic function. If you input “2”, the backtrack approach will be executed using the minimum remaining values (MVR) heuristic and least constraint value heuristic. If you input “3”, the cost (optional part) will be considered.
   5. For the optional part, the file name of cost should be cost.txt.
2. **Approach**
   1. AC3

AC3 is used in backtracking, maintaining the arc consistency every time a value was assigned.

*Function AC-3(csp) returns false if an inconsistency is found and true otherwise*

*Inputs: csp*

*Local variables: queue(initially all the arcs in constraints)*

*While queue is not empty do*

*(Xi,Xj)=queue.pop()*

*If Revise(csp,Xi,Xj) then*

*If size of Di=0 then return false*

*For each Xk in Xi.Neighbors-{Xj} do*

*Add(Xk,Xi)to queue*

*Return true*

*Function Revise(csp,Xi,Xj) returns true iff revise the domain of Xi*

*Revise=false*

*For each x in Di do*

*If no value y in Dj allows(x,y) to satisfy the constraint between Xi and Xj then*

*Delete x from Di*

*Revised=true*

*Return revised*

* 1. Backtracking

Backtracking is the main way of researching. In this approach, minimum remain value heuristic is used to choose the variable. Least constraint value heuristic is used to choose the value that will be assigned to the variable. And for each assignment, AC3 is used to inference the availability.

*Function Backtracking\_search(csp) returns a solution or failure*

*Function Backtrack(assignment):*

*If assignment is complete return assignment*

*Var=select\_unassigned\_variable(assignment, csp)*

*(using MRV)*

*For values in Order\_domain\_values(var,assignment,csp)*

*(using LCV)*

*If value is consistent with assignment then*

*Csp.assign(var, value, assignment)*

*Remove value from csp.current\_domain*

*If inference !=failure(using AC3 as inference)*

*Result=Backtrack(assignment)*

*If result is not none:*

*Return result*

*Return none*

* 1. Minimum remain value heuristic

*Function mrv\_degree(assignment, csp):*

*Dictionary[csp.variable]=variable.current\_domain*

*Order=Sort Dictionary by the quantities of items*

*While quantities of variables:*

*Variable=the first variable in order*

*If there are two variables has the same remain value*

*Variable= the lenth is the largest*

*Return variable*

* 1. Least constraint value heuristic

Choose the value that most frequently occurring value in the constraints.

* 1. Maxlength heuristic

The variable whose length is the largest and will be chosen first to be assigned to optimal the cost.

* 1. Minimum cost heuristic

The value that has the smallest cost will be first assigned to the variable.

1. **Result**

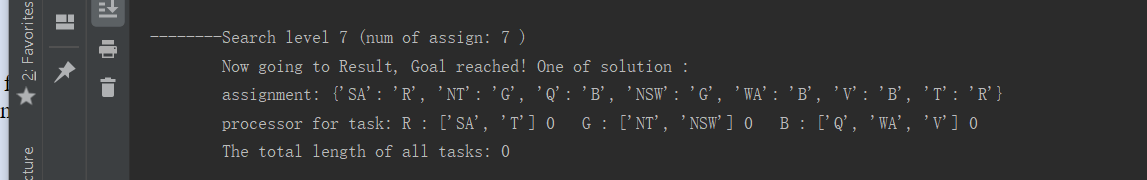
We tried many test file for our project, also Australian map coloring problem. Our program performs pretty, all tests are passed. We also solved the optional problem.

Fig1. the result of the Australian map coloring problem

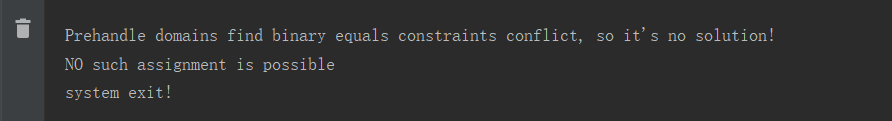
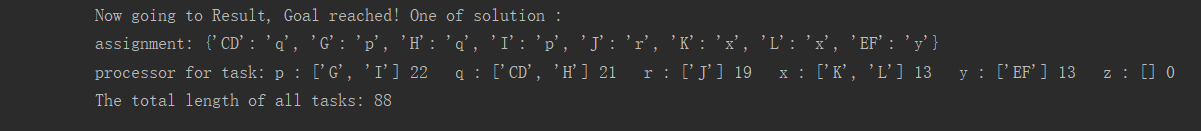


Fig2. the result of input1.txt on the assignment website

Fig3. The result of input2.txt on the assignment website

1. **Strength and weakness**

The strengths of our program are the heuristic function is uncoupled with the CSP class, which means you could alert the heuristic function without changing other parts of the code, that’s what we use to solve the optional problem. You just need to change the heuristic function so that the sort of selecting variables would different.

The weakness of our program is, for now, we can’t take arithmetic constraints into consideration, so we are failed when we tried to test the cryptarithmetic. But we don’t think that’s unsolvable, we just need more time.