CSE331 Assignment Report

Ahmed Semih Özmekik

In this report, necessary and sufficient explanations about the assignment will be given with some screenshots.

Bonus Part

In the assignment file, reading the sets from the file is stated as an bonus score; regarding to that, as can be seen in the first section, this bonus part is made, ie the clusters are read from the file in my implementation.

Rule Set

Before hand, it is necessary to specify the simple rules I set regarding the file reading. Syntax can be seen from any instance of given input files:

```
1,2,3,4,5
4,1,3
2,5
1,4,3,2
;
```

First line represents, set *X* and ongoing lines represents respectively, S_0, S_1, S_2, ..., S_m which are being subsets of *X*.[1] As can be seen, the elements are separated by commas and each set is shown on a different line and a semicolon is added at the bottom with a new line after the desired entries are given. There shouldn't be any other characters in the file after the semicolon, meaning any return or in particular the new line character.[2]

Because the violation of these rules is not considered an exception in the program, user may lead to different and incorrect results or run time crashes, worstly non-terminating programs; in short violation of these rules will not be given to user as warning or error, they are simply undefined behaviours.

No natural numbers other than single-digit numbers are accepted. All other characters in *ascii* can be used to represent more and different elements. Samples can be seen in some other test files.

1 Based on the definitions in the assignment. Accordingly, *S* sets are subsets of *X* set.
2 *i.e. '\n'*

Code Explanation

If the code explanation will be elaborated in detail, the volume of this report will beyond it's volume, so relying on the comments contained in the code, I believe they can be understood at a certain level. Therefore, codes will not be disclosed. Rather instead, we directly focus on test and results.

Test Cases

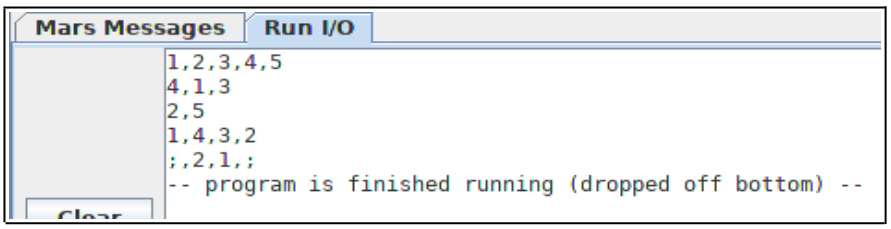
I. Input file:

```
1,2,3,4,5
4,1,3
2,5
1,4,3,2
;
```

X = { 1, 2, 3, 4, 5 } S_0 = { 1, 2, 3, 4, 5 } S_1 = { 1, 2, 3, 4, 5 } S_2 = { 1, 2, 3, 4, 5 }

I = { 1, 2 }

Output:



II. Input File[3]:

```
1,2,3,4,5,6,7,8,9,A,B,C,D
1,2
2,3,4,5
6,7,8,9,A,B,C,D
1,3,5,7,9,B,D
2,4,6,8,A,C,D
;
```

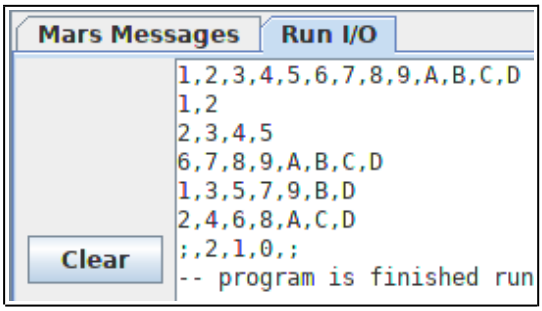
X = { 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D } S_0 = { 1, 2} S_1 = { 2, 3, 4, 5 }

I = { 0, 1, 2} S_2 = { 6, 7, 8, 9, A, B, C, D }

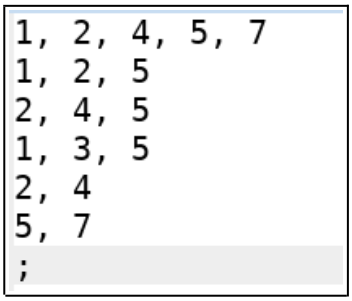
S_3 = { 1, 3, 5, 7, 9, B, D } S_4 = { 2, 4, 6, 8, A, C, D }

Output:

3 The optimal solution for this file must be I = { 3, 4 }. Yet the greedy algorithm must produce the result I = { 0, 1, 2 }.



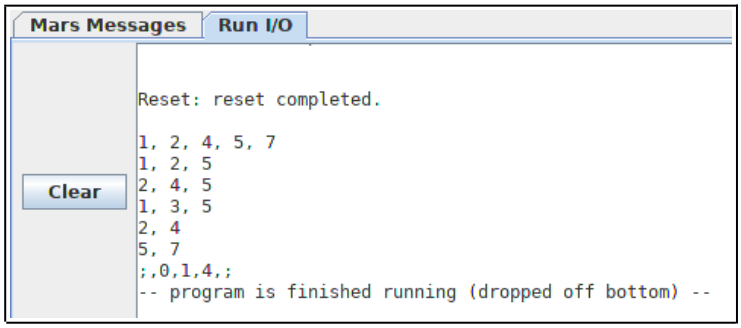
III. Input File:



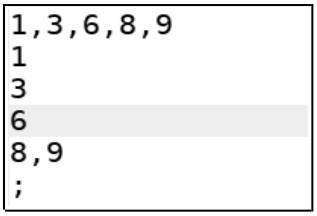
X **= { 1, 2, 4, 5, 7}** ***S₀*** **= { 1, 2, 5}** ***S₁*** **= { 2, 4, 5 }** ***S₂*** **= { 1, 3, 5}**

I = { 0, 1, 4 } ***S₃*** **= { 2, 4 }** ***S₄*** **= { 5, 7 }**

Output:



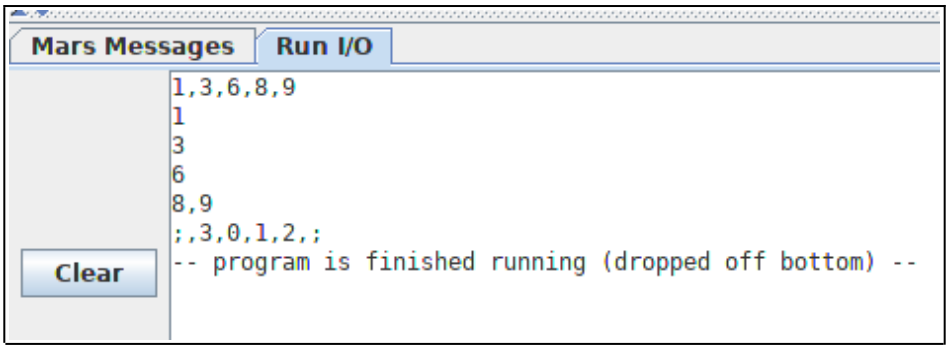
IV. Input File:



X = { 1, 3, 6, 8, 9} **S_0 = { 1 }** **S_1 = { 3 }** **S_2 = { 6}** **S_3 = { 8, 9}**

I = { 0, 1, 2, 3 }

Output:



Testing Notes

Those are changable things of my program, yet there are still such parts that hardcoded. Only input file are considered to be changable. In the existence of big sets, space stating data statements might be slightly changed.

And of course, input files must be in the same folder with Mars4_5.jar executable.

```
.data
inputfile: .asciiz "input2.txt"
buffer: .space 128      # Buffer size can be modified regarding to input
delimeter: .ascii "\n"
endfile:  .ascii ";"
removed: .ascii "X"     # Junk value to represent removed item

I: .space 32            # Resultant
X: .word 0              # Starting address of X
S: .word 0              # Starting address of sets
```