Natural Computing

Assignment 3

PARTH KANAKIYA

Id - 201410100

Q.1)

Binary Counter:

Gmc	Gse	Ratio(Gmc/Gse)	Errors
3	9	0.33	4823
3	6	0.5	1335
6	6	1	1269
6	2	3	No reaction
17	8.6	1.9767	53

Sierpinski 2x2 Tiles:

Gmc	Gse	Ratio(Gmc/Gse)	Errors
17	8.6	1.9767	0
17	4.3	3.9534	No reaction
8.6	8.6	1	5660
4.3	17	0.2529411765	8432
8.6	17	0.5058823529	18104

Observations:

- The ratio Gmc/Gse is equal to threshold t.That is, it is ratio of concentration of tiles to strength of individual bonds.
- When Gse is large and Gmc is small tiles have a higher tendency to associate and lower tendency to dissociate. In region $0<t<1$, aggregates grow quickly but have greater error.
- In contrast,when Gse is small and Gmc is large, tiles have a lower tendency to associate and higher tendency to dissociate. Hence, aggregates do not grow in $t>2$.
- Optimal growth is realized when t is close to 2, implying we must sacrifice growth speed to maintain a low error rate.

Q.2)For tile sets Binary Counter Square.tiles, explain the .tile file and also find the functions if any and describe the tile set.

Solution:

Parameter	Description
tile edges matches {{N E S W}*}	Depicts encoding sequence of the .tile file
num binding types	Total different binding types.
tile edges	Edges encoded in {N E S W} format is represented by this parameter.
binding strengths	The binding strength which is a parameter representing the association/binding strength is to be set by this.
block	How much the size of the window must be is to be specified in this parameter.
Gse,Gmc	Command line parameter for setting the value of 'Gmc' and 'Gse', and The ratio Gmc/Gse is equal to threshold t.That is, it is ratio of concentration of tiles to strength of individual bonds.
num tile types	Total number of different tiles.

- tile edges matches {{N E S W}*} -> Orientation of the tiles.
- num tile types=28 -> Total number of different tiles

- num binding types=23 -> Tiles except computational tiles, binding tiles, seed,boundaries
- tile edges={

{15 1 8 0} (red)
{6 2 8 1} (red4)
{5 3 8 2} (red3)
{5 4 17 3} (red2)
{19 7 7 4} (red1)
{17 7 7 9}
{8 9 17 8}
{7 7 7 7}[2] (pink1)
{8 8 8 8}[2] (pink2)
{9 7 7 18}
{8 18 9 8}
{5 12 5 12} (blue3)
{6 12 6 12} (green)
{5 10 6 10} (blue2)
{6 10 5 11} (green2)
{6 11 6 11} (green3)
{5 11 5 11} (blue1)
{20 8 13 12}(blue4)
{21 8 14 12}(green4)
{13 18 19 10}(blue4)
{13 8 21 10}(blue4)
{14 8 20 11}(green4)
{15 12 22 0} (darkcyan)
{16 12 23 0} (lightgreen)
{0 10 16 0}
{23 10 15 0} (lime)
{22 11 15 0} (blue3)
{23 11 16 0} (yellow)
}

All the tiles. Here numbers represent their sides, in N E S W order.

- binding strengths=

  {2 2 2 2 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 }

  Binding strength of the binding tiles. To make sure where to stop computation, the tiles should not grow etc.

- size=64
- block=10
- seed=53,5,1

Initial seed

- update_rate=5000
- Gse=12.2

  Gse(interaction free energy per binding)

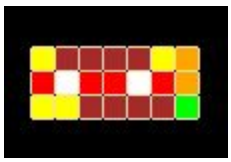
- Gmc=24

Gmc(initiation free energy (units kT))

Q.3) Download and install the software http://www.guptalab.org/xtilemod/. Try to generate tile set for doing different arithmetic operations of 2 integers and n integers. Try all options available in the software and after generating the .tiles file simulate it using xgrow. Explain one of them completely. Do the same for primality testing.

Solution:

Adding 15 and 30 using 8 tile adder:



Colouring Pattern:

Seed Tile : Green

Output Tile: Red = 1 White = 0

Input Tile: Brown = 1 yellow = 0

Inputs:

1st row in image represents number 15 in binary: 001111

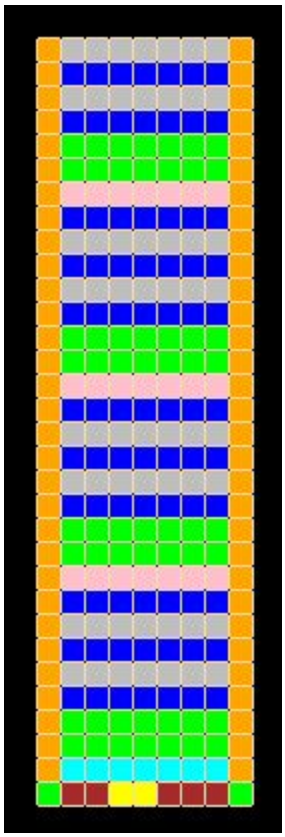
3rd row in image represents number 30 in binary: 011110

Output:

2nd row represents number 45 in binary : 101101

Primality testing:

The input number is taken to be 103:



Coloring Pattern:

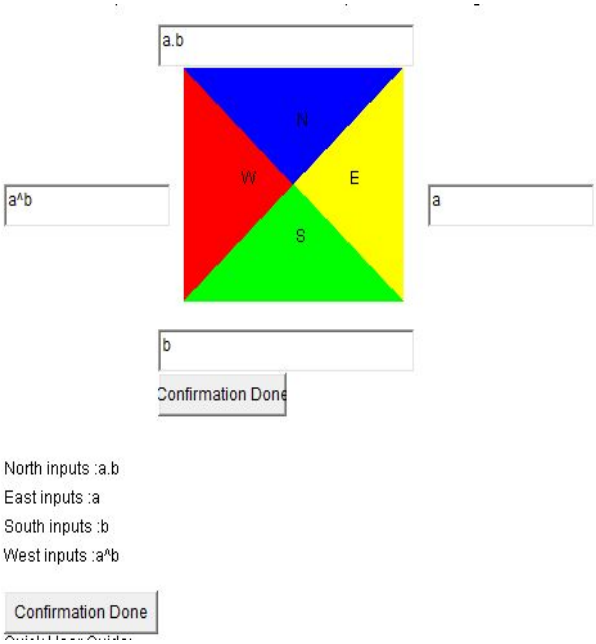
Red Color: Means Number is Composite

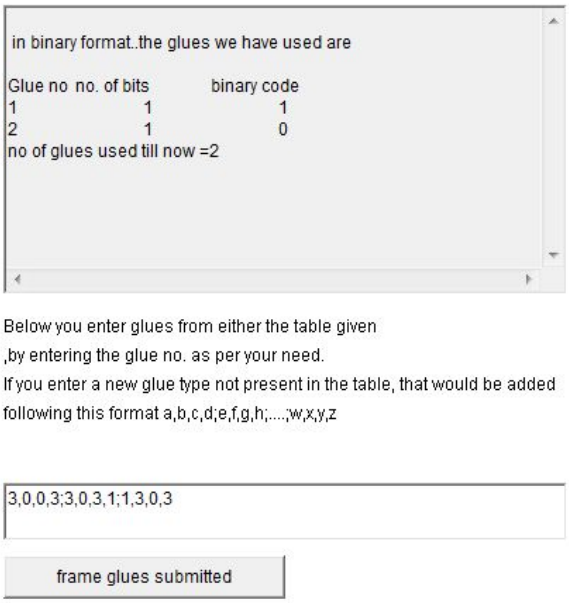
Grey Color: Means Number is Prime

- Top most row signifies the output. The grey color signifies that it is a prime number.

Q5) Run the software Xtile 1.0 at http://www.guptalab.org/xtile. Generate some .tiles files and attach them

FIRST TILE GENERATED FOR FOLLOWING TILE CONFIGURATION





OUTPUT FOR 1st CONFIGURATION ABOVE

/*This .tiles file was generated using XTile 1.0 at 2017-02-27 23:24:04

developed by Anshul Chaurasia, Sudhanshu Dwivedi and Prateek Jain, DA-IICT,INDIA

detailed documentation and tool available at www.guptalab.org/xtile

this .tiles file is input to Xgrow developed by Eric Winfree, Caltech */

tile edges matches {{N E S W}*}

num tile types=7

num binding types=3

tile edges={

{3 0 0 3}

{3 0 3 1}

{1 3 0 3}

{2 2 2 2}

{2 2 1 1}

{2 1 2 1}

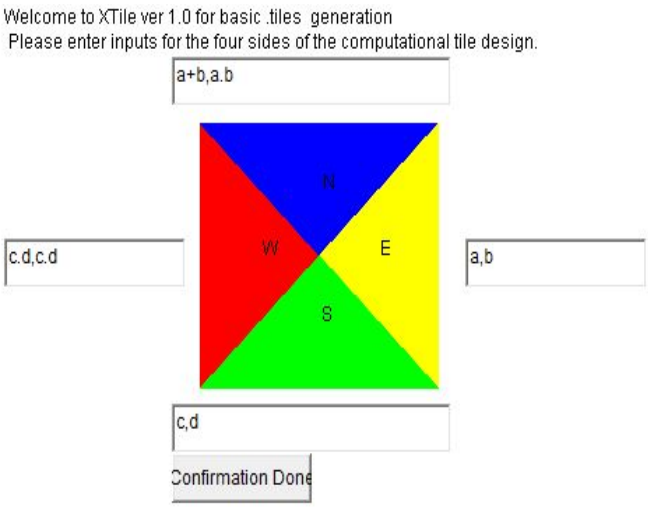
{1 1 1 2}

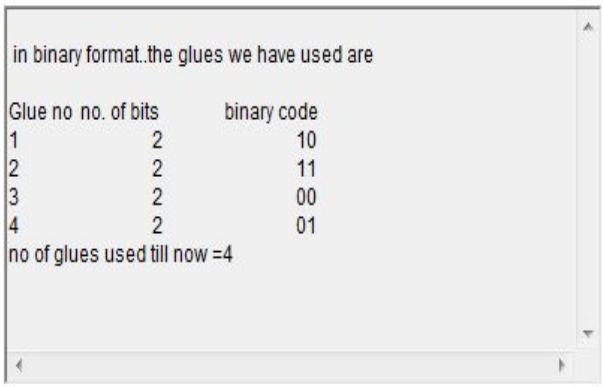
}

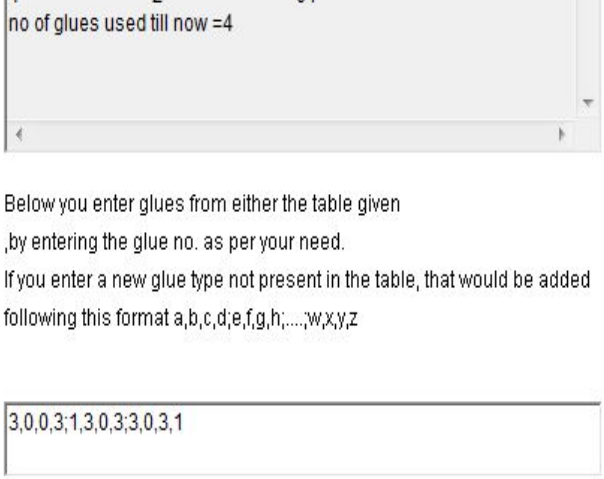
binding strengths=

{1 1 2 }

SECOND TILE FILE GENERATED FOR FOLLOWING INPUT TILE CONFIGURATION







OUTPUT FOR 2nd CONFIGURATION ABOVE

/*This .tiles file was generated using XTile 1.0 at 2017-02-27 23:29:56

developed by Anshul Chaurasia, Sudhanshu Dwivedi and Prateek Jain, DA-IICT,INDIA

detailed documentation and tool available at www.guptalab.org/xtile

this .tiles file is input to Xgrow developed by Eric Winfree, Caltech */

tile edges matches {{N E S W}*}

num tile types=19

num binding types=4

tile edges={

{3 0 0 3}

{1 3 0 3}

{3 0 3 1}

{3 3 3 3}

{3 3 4 3}

{3 3 1 3}

{3 3 2 2}

{1 4 3 3}

{1 4 4 3}

{1 4 1 3}

{1 4 2 2}

{1 1 3 3}

{1 1 4 3}

{1 1 1 3}

{1 1 2 2}

{2 2 3 3}

{2 2 4 3}

{2 2 1 3}

{2 2 2 2}

}

binding strengths=

{1 1 1 1 }

PROOF THAT I HAVE DONE THE WORK