Tyler Tracey

Chip Klostermeyer

CIS 4560

**Dominating Sets and Roman-Weighted Dominating Sets**

**Overview:**

So my program does almost exactly what specified. I let the user input how many graphs they would like to generate. Then I generate a random edge density between 10 and 80, and a random number of vertices between 12 and 30, for each graph. I then pass the graphs along and find the size of the smallest dominating set, and then I use that to find the weight of the lowest-weight roman dominating set.

**Algorithms**

For the dominating set, I think my algorithm is, although not generic by any means, probably not very eloquent. I loop through every vertex as my “starting vertex.” If that vertex I am working on is not dominated, I put it in the dominating set and set all of its neighbors to the dominated state. If the vertex is dominated I do nothing. I find the sequential domination size starting from each vertex in that fashion. The algorithm was basically just designed as way to 100% of the time find the size of the smallest dominating set, without being the exhaustive search.

I based my algorithm for the roman domination off of the smallest dominating set I found in the previous section. If a vertex is in the dominating set I weight it as a 2. If it is not, I weight it as a 0. Then I loop through the vertices weighted as 2 and tried to find what I call “useless 2’s.” If a 2 is not connected to any 0’s that are not, in turn, connected to at least one other 2, I make that 2 a 1-vertex. To be clearer, I take any vertex weighted 2, and see if any of it neighbors weighted 0 NEED that vertex to be weighted 2. If not, I make it a 1. Afterwards I loop through again and check if there is any vertex weight with a 1 that is connected to a 2. If it is, I change its weight to 0.

**Results**

I am 100% confident that my program always find the smallest possible dominating set, as well as the smallest possible roman domination weight. However, I find one thing really curious. In the specifications for the assignment you said to pay close attention to how edge density affects the percentage of graphs that have domination number half the roman domination weight. In my testing, for graphs with between 12 and 30 vertices, there is no edge density probability, between 5 and 80, that causes a graph to *not* have domination number exactly half the roman domination number. I think it is possible to stray from this with smaller graphs, but in my testing every single graph had domination number exactly half the roman domination number. I would also like to mention that I did not disregard graphs that are not connected (in other words, I did not do the extra credit).

**Time**

This program actually runs pretty fast. For one graph to be analyzed it takes about 0.17 seconds on average. For 100 graphs to be analyzed it takes about 0.34 seconds. In fact, even 10,000 graphs does not push the program to take even half a second. So, in that area, I think the program does pretty well.

**Sample Output**

**This output is from the analysis of 20 graphs. I will also include a file that has output from the analysis of 1000 graphs (just in case you want more), but it will be separate from this because of the sheer volume of output due to how I organized my output.**

Number of graphs to make? 20

Graph 0 NumVerts: 26 EdgeProb: 41

Graph 1 NumVerts: 18 EdgeProb: 15

Graph 2 NumVerts: 18 EdgeProb: 55

Graph 3 NumVerts: 19 EdgeProb: 66

Graph 4 NumVerts: 12 EdgeProb: 36

Graph 5 NumVerts: 21 EdgeProb: 18

Graph 6 NumVerts: 29 EdgeProb: 72

Graph 7 NumVerts: 26 EdgeProb: 24

Graph 8 NumVerts: 23 EdgeProb: 58

Graph 9 NumVerts: 29 EdgeProb: 61

Graph 10 NumVerts: 21 EdgeProb: 54

Graph 11 NumVerts: 27 EdgeProb: 51

Graph 12 NumVerts: 24 EdgeProb: 44

Graph 13 NumVerts: 23 EdgeProb: 16

Graph 14 NumVerts: 24 EdgeProb: 65

Graph 15 NumVerts: 24 EdgeProb: 25

Graph 16 NumVerts: 15 EdgeProb: 51

Graph 17 NumVerts: 18 EdgeProb: 41

Graph 18 NumVerts: 18 EdgeProb: 38

Graph 19 NumVerts: 29 EdgeProb: 32

Minimum dominating set size for graph 0 is 5

Minimum roman weighting for graph 0 is 10

Minimum dominating set size for graph 1 is 9

Minimum roman weighting for graph 1 is 18

Minimum dominating set size for graph 2 is 3

Minimum roman weighting for graph 2 is 6

Minimum dominating set size for graph 3 is 2

Minimum roman weighting for graph 3 is 4

Minimum dominating set size for graph 4 is 2

Minimum roman weighting for graph 4 is 4

Minimum dominating set size for graph 5 is 8

Minimum roman weighting for graph 5 is 16

Minimum dominating set size for graph 6 is 2

Minimum roman weighting for graph 6 is 4

Minimum dominating set size for graph 7 is 6

Minimum roman weighting for graph 7 is 12

Minimum dominating set size for graph 8 is 3

Minimum roman weighting for graph 8 is 6

Minimum dominating set size for graph 9 is 3

Minimum roman weighting for graph 9 is 6

Minimum dominating set size for graph 10 is 3

Minimum roman weighting for graph 10 is 6

Minimum dominating set size for graph 11 is 3

Minimum roman weighting for graph 11 is 6

Minimum dominating set size for graph 12 is 3

Minimum roman weighting for graph 12 is 6

Minimum dominating set size for graph 13 is 9

Minimum roman weighting for graph 13 is 18

Minimum dominating set size for graph 14 is 2

Minimum roman weighting for graph 14 is 4

Minimum dominating set size for graph 15 is 6

Minimum roman weighting for graph 15 is 12

Minimum dominating set size for graph 16 is 3

Minimum roman weighting for graph 16 is 6

Minimum dominating set size for graph 17 is 3

Minimum roman weighting for graph 17 is 6

Minimum dominating set size for graph 18 is 4

Minimum roman weighting for graph 18 is 8

Minimum dominating set size for graph 19 is 5

Minimum roman weighting for graph 19 is 10

The percentage of graphs with dom number half or less of roman weight is 100.0%

Time to finish testing: 0.1591 seconds