**IT 328, Program 1  
Students:**

Rachel Schifano, raschif  
Ben Gulans, bmgulan

Unix Account, raschif

Password, spring14

Location: raschif/IT328/program1

**Summary**

In order to approach solving this assignment, we understood it was important to complete the Clique class first. Determining cliques was solvable by using an ArrayList to store our vertices and check for the max clique that would result from a graph. It was apparent that both the Clique and Independent Sets would require reading in graphs that were similar, so any basic file reading was not necessarily challenging. However, when solving the Independent Sets, it was a little more complicated, until we realized we could solve for the complement of the current graph in order to determine those vertices that did not have edges or connections to each other. In general, the first two programs were related and, although mildly time consuming, they were manageable.

However, it was solving the CNF class that proved the most difficult considering many challenges we faced. We ideally wanted to program with good object oriented and modularizing design in mind, but time and our desperation to complete these tasks among our other courses was pressing. Using static classes and running majority of our graph scanning row by row through the main gave us very messy code to work with. We knew we had to use our Clique program to help solve the CNF, so we played with some global variables and our code is very disorganized. We struggled with the logic for converting the CNF to a graph that can be analyzed through the Clique class. We wrote an output file to visually see the graphs that the CNF were reduced to, and realized the vertex and edge values were off. Initially, we thought it was our logic when reducing the CNF. We experimented, playing with the parity bits, questioning our logic for converting a CNF relationship to a 1 or 0 in a graph, and even considered our Clique class was wrong (even though, it was correctly solving Cliques). It turned out that we were passing in an ArrayList that did not clear its current values, and thus was compounding CNF equations. It was the fact that each graph looked the same for the first 12 values, as a result of still storing CNF values for problem 1, looked exactly the same. We were stuck on this bug for two hours before fixing it. Adding a simple cnf.clear() to the end of our while loop solved it. The messy nature of our code also resulted in us struggling to keep track of the proper ArrayList storing the actual max clique indices that corresponded to the CNF equation. The convoluted nature to the way we stored these critical values made it hard to easily print out the results with proper assignments.

In summary, the program was challenging but fun. It really helped reinforce learning about Cliques, Independent Sets, and 3SAT for 3CNF since we had to repeatedly take notes and trace input/output through the logic of solving these problems to ensure our program was solving them correctly.

**Output**

Clique

RAVINE:program1 schifano$ java Clique

\* Max Cliques in graphs in graphsDense.txt

(|V|,|E|) Cliques (size, ms used)

G1 (2, 2) [0](size=1, 0 ms)

G2 (2, 3) [0, 1](size=2, 1 ms)

G3 (4, 8) [1, 2, 3](size=3, 0 ms)

G4 (4, 7) [0, 3](size=2, 0 ms)

G5 (6, 9) [0, 4](size=2, 0 ms)

G6 (6, 16) [0, 2, 3, 5](size=4, 0 ms)

G7 (8, 17) [1, 3, 6](size=3, 0 ms)

G8 (8, 29) [0, 2, 3, 6](size=4, 2 ms)

G9 (10, 31) [5, 6, 8, 9](size=4, 0 ms)

G10 (10, 49) [0, 1, 3, 5, 8, 9](size=6, 0 ms)

G11 (12, 43) [2, 4, 8, 10](size=4, 0 ms)

G12 (12, 68) [1, 3, 4, 5, 6, 7, 10, 11](size=8, 1 ms)

G13 (14, 47) [1, 5, 7, 10](size=4, 0 ms)

G14 (14, 93) [0, 2, 4, 5, 6, 7, 10, 11, 13](size=9, 1 ms)

G15 (16, 54) [0, 3, 8](size=3, 0 ms)

G16 (16, 115) [0, 1, 4, 5, 6, 7, 8, 10, 12](size=9, 2 ms)

G17 (18, 82) [0, 1, 6, 9](size=4, 0 ms)

G18 (18, 133) [1, 2, 5, 9, 10, 13, 14, 16](size=8, 2 ms)

G19 (20, 105) [0, 1, 2, 6, 13](size=5, 0 ms)

G20 (20, 173) [3, 5, 6, 8, 9, 12, 14, 17, 18, 19](size=10, 4 ms)

G21 (22, 122) [0, 1, 4, 7](size=4, 0 ms)

G22 (22, 202) [1, 4, 5, 6, 11, 12, 13, 17, 18, 19](size=10, 4 ms)

G23 (24, 129) [1, 2, 8, 11](size=4, 0 ms)

G24 (24, 243) [0, 2, 3, 6, 7, 9, 10, 13, 16](size=9, 10 ms)

G25 (26, 152) [4, 5, 17, 20, 25](size=5, 1 ms)

G26 (26, 273) [0, 1, 2, 5, 9, 10, 12, 17, 22](size=9, 11 ms)

G27 (28, 175) [0, 4, 10, 12, 13, 17](size=6, 1 ms)

G28 (28, 322) [1, 2, 3, 12, 14, 15, 17, 18, 21, 22, 25, 26, 27](size=13, 12 ms)

G29 (30, 194) [4, 6, 14, 17, 23](size=5, 0 ms)

G30 (30, 380) [1, 2, 3, 7, 8, 11, 12, 17, 20, 24, 25, 28](size=12, 21 ms)

G31 (32, 228) [0, 7, 11, 20, 29](size=5, 0 ms)

G32 (32, 451) [1, 3, 4, 5, 6, 7, 10, 11, 13, 15, 16, 20, 27](size=13, 75 ms)

G33 (34, 275) [0, 1, 10, 14, 27](size=5, 1 ms)

G34 (34, 476) [0, 2, 3, 5, 7, 11, 12, 14, 16, 26, 27, 28, 32](size=13, 37 ms)

G35 (36, 308) [1, 6, 9, 21, 28, 31](size=6, 0 ms)

G36 (36, 535) [0, 6, 8, 9, 14, 19, 20, 22, 26, 27, 34, 35](size=12, 65 ms)

G37 (38, 331) [3, 10, 18, 21, 25, 34](size=6, 0 ms)

G38 (38, 596) [0, 1, 3, 5, 6, 8, 10, 11, 17, 19, 30, 34, 35](size=13, 86 ms)

G39 (40, 373) [1, 7, 12, 15, 28, 37](size=6, 1 ms)

G40 (40, 673) [0, 2, 9, 10, 14, 20, 22, 23, 25, 35, 36, 37, 39](size=13, 198 ms)

G41 (42, 388) [2, 6, 7, 11, 20, 23](size=6, 1 ms)

G42 (42, 737) [3, 10, 11, 15, 21, 23, 24, 27, 28, 29, 31, 34, 35, 38](size=14, 252 ms)

G43 (44, 407) [0, 1, 7, 21, 34, 36](size=6, 1 ms)

G44 (44, 791) [0, 1, 2, 3, 15, 17, 18, 23, 29, 34, 35, 39, 41](size=13, 216 ms)

G45 (46, 439) [9, 16, 25, 31, 41, 45](size=6, 0 ms)

G46 (46, 862) [0, 3, 5, 6, 8, 9, 12, 14, 16, 28, 33, 37, 39, 44](size=14, 437 ms)

G47 (48, 514) [7, 11, 13, 15, 23, 26, 38](size=7, 2 ms)

G48 (48, 962) [0, 2, 7, 8, 14, 21, 26, 29, 32, 36, 37, 40, 41, 45, 47](size=15, 1073 ms)

G49 (50, 538) [0, 9, 11, 16, 31, 43](size=6, 2 ms)

G50 (50, 1015) [1, 2, 8, 9, 12, 16, 19, 21, 31, 32, 38, 40, 44, 46](size=14, 699 ms)

G51 (52, 593) [0, 14, 21, 26, 29, 34, 39](size=7, 2 ms)

G52 (52, 1089) [1, 2, 3, 13, 18, 22, 27, 28, 31, 32, 37, 41, 46, 47](size=14, 1238 ms)

G53 (54, 647) [5, 9, 10, 11, 19, 37, 39, 41](size=8, 2 ms)

G54 (54, 1181) [3, 10, 12, 14, 23, 27, 28, 29, 31, 32, 37, 40, 48, 50](size=14, 1295 ms)

G55 (56, 668) [1, 15, 22, 28, 29, 33, 35](size=7, 4 ms)

G56 (56, 1261) [1, 4, 12, 19, 21, 22, 24, 26, 27, 30, 32, 35, 39, 45, 50, 53](size=16, 2101 ms)

G57 (58, 718) [10, 19, 21, 22, 27, 31, 51](size=7, 2 ms)

G58 (58, 1368) [0, 3, 12, 15, 18, 25, 28, 29, 37, 42, 43, 47, 49, 51, 56](size=15, 3505 ms)

G59 (60, 750) [3, 4, 10, 32, 33, 36, 48](size=7, 2 ms)

G60 (60, 1463) [11, 13, 14, 17, 19, 20, 26, 32, 35, 37, 43, 44, 47, 48, 51, 56](size=16, 5062 ms)

Independent Set

RAVINE:program1 schifano$ java IndependentSet

\* Max Independent Set in graphs in graphs.txt

(|V|,|E|) Independent Sets (size, ms used)

G1 (2, 1) [0, 1](size=2, 0 ms)

G2 (2, 1) [0, 1](size=2, 0 ms)

G3 (4, 5) [0, 1, 2](size=3, 0 ms)

G4 (4, 5) [0, 1, 3](size=3, 0 ms)

G5 (6, 13) [0, 1, 2, 5](size=4, 0 ms)

G6 (6, 10) [1, 2, 3, 4](size=4, 0 ms)

G7 (8, 22) [0, 1, 2, 3, 6](size=5, 1 ms)

G8 (8, 14) [0, 5, 6, 7](size=4, 3 ms)

G9 (10, 37) [0, 3, 4, 5, 7, 9](size=6, 1 ms)

G10 (10, 32) [1, 3, 5, 6, 7](size=5, 0 ms)

G11 (12, 55) [0, 2, 4, 5, 6, 8, 9, 10](size=8, 2 ms)

G12 (12, 43) [0, 4, 5, 9, 10](size=5, 1 ms)

G13 (14, 76) [1, 3, 4, 5, 7, 9, 11](size=7, 1 ms)

G14 (14, 54) [0, 2, 5, 7, 13](size=5, 0 ms)

G15 (16, 99) [0, 1, 2, 3, 6, 10, 11, 13](size=8, 2 ms)

G16 (16, 67) [0, 1, 3, 5, 13](size=5, 0 ms)

G17 (18, 117) [1, 4, 5, 6, 10, 12, 16, 17](size=8, 2 ms)

G18 (18, 99) [1, 3, 5, 10, 11, 15](size=6, 0 ms)

G19 (20, 150) [0, 1, 2, 3, 7, 10, 14, 19](size=8, 3 ms)

G20 (20, 127) [1, 6, 7, 10, 11, 13, 15, 17, 19](size=9, 1 ms)

G21 (22, 188) [0, 1, 2, 5, 6, 7, 9, 12, 17, 18](size=10, 10 ms)

G22 (22, 140) [0, 3, 6, 7, 8, 9, 13, 16](size=8, 2 ms)

G23 (24, 225) [1, 3, 7, 8, 9, 12, 17, 19, 20, 22](size=10, 12 ms)

G24 (24, 161) [0, 2, 12, 13, 20, 22](size=6, 1 ms)

G25 (26, 261) [1, 2, 5, 6, 7, 9, 10, 14, 17, 22, 24](size=11, 15 ms)

G26 (26, 199) [0, 2, 3, 4, 17, 21, 24](size=7, 0 ms)

G27 (28, 311) [0, 1, 4, 6, 7, 10, 12, 14, 15, 16, 20](size=11, 14 ms)

G28 (28, 246) [0, 3, 6, 9, 14, 17, 18, 19](size=8, 2 ms)

G29 (30, 344) [2, 3, 4, 6, 9, 10, 19, 22, 23, 25, 26, 29](size=12, 18 ms)

G30 (30, 260) [0, 4, 5, 6, 8, 9, 12](size=7, 1 ms)

G31 (32, 386) [2, 3, 7, 8, 10, 12, 13, 14, 20, 21, 22, 23](size=12, 22 ms)

G32 (32, 302) [0, 4, 8, 11, 16, 17, 22, 27](size=8, 1 ms)

G33 (34, 449) [0, 2, 4, 6, 7, 8, 15, 16, 18, 23, 27, 31, 32](size=13, 37 ms)

G34 (34, 329) [4, 9, 15, 20, 22, 23, 27, 29](size=8, 2 ms)

G35 (36, 503) [0, 5, 6, 10, 13, 17, 20, 22, 24, 25, 27, 32, 33](size=13, 50 ms)

G36 (36, 384) [0, 3, 4, 7, 11, 13, 14, 23](size=8, 4 ms)

G37 (38, 551) [0, 1, 3, 5, 7, 10, 14, 21, 23, 25, 26, 30](size=12, 58 ms)

G38 (38, 437) [0, 2, 5, 13, 19, 23, 28, 30, 34](size=9, 2 ms)

G39 (40, 607) [0, 1, 5, 6, 10, 12, 16, 21, 31, 33, 36, 37, 38](size=13, 90 ms)

G40 (40, 464) [1, 7, 13, 15, 19, 22, 27](size=7, 4 ms)

G41 (42, 690) [3, 4, 5, 7, 8, 11, 17, 21, 26, 27, 28, 29, 32](size=13, 160 ms)

G42 (42, 521) [2, 3, 9, 13, 18, 24, 25, 28, 37](size=9, 4 ms)

G43 (44, 760) [1, 2, 5, 6, 7, 11, 12, 21, 23, 25, 26, 28, 35, 39](size=14, 347 ms)

G44 (44, 581) [0, 2, 4, 8, 17, 22, 39, 40, 42](size=9, 6 ms)

G45 (46, 847) [0, 2, 5, 9, 11, 12, 16, 17, 24, 26, 27, 28, 38, 41](size=14, 647 ms)

G46 (46, 603) [1, 4, 7, 12, 16, 18, 32, 33, 43](size=9, 5 ms)

G47 (48, 911) [8, 10, 11, 14, 15, 16, 22, 24, 25, 27, 31, 33, 34, 35, 46](size=15, 792 ms)

G48 (48, 643) [5, 10, 15, 16, 22, 31, 35, 43](size=8, 4 ms)

G49 (50, 963) [4, 6, 8, 15, 16, 17, 19, 21, 29, 30, 31, 33, 34, 47, 49](size=15, 805 ms)

G50 (50, 716) [1, 2, 4, 10, 20, 24, 31, 33, 37](size=9, 9 ms)

G51 (52, 1057) [0, 3, 12, 21, 22, 27, 28, 30, 31, 37, 38, 41, 45, 47, 50](size=15, 1181 ms)

G52 (52, 813) [0, 2, 3, 6, 15, 17, 24, 36, 44](size=9, 18 ms)

G53 (54, 1136) [2, 3, 5, 11, 16, 17, 18, 19, 24, 25, 29, 30, 35, 43, 52](size=15, 1510 ms)

G54 (54, 830) [2, 9, 16, 21, 23, 25, 35, 50, 51](size=9, 10 ms)

G55 (56, 1246) [0, 3, 5, 6, 8, 9, 13, 14, 27, 30, 32, 45, 46, 48, 51](size=15, 3136 ms)

G56 (56, 912) [0, 5, 12, 13, 14, 27, 39, 41, 51](size=9, 18 ms)

G57 (58, 1311) [0, 6, 13, 17, 18, 20, 23, 27, 32, 33, 37, 43, 47, 54, 57](size=15, 2925 ms)

G58 (58, 1018) [1, 6, 16, 27, 30, 31, 37, 45, 57](size=9, 24 ms)

G59 (60, 1428) [1, 6, 10, 11, 12, 17, 19, 22, 23, 31, 33, 36, 38, 41, 44, 48, 49, 55](size=18, 6024 ms)

G60 (60, 1068) [0, 4, 7, 8, 11, 24, 27, 36, 52, 57](size=10, 28 ms)

CNF – Note, the output is a result of running the program for more than 10 minutes.

RAVINE:program1 schifano$ java CNF

3-CNF No.1: [n=3 k=4] Assignments: [ A1=F A2=F ] (0 ms)

3-CNF No.2: [n=3 k=4] Assignments: [ A1=T A2=T A3=T ] (3 ms)

3-CNF No.3: [n=3 k=8] Assignments: [ A1=F A2=T A3=T ] (5 ms)

3-CNF No.4: [n=3 k=8] Assignments: [ A1=F A2=T A3=T ] (18 ms)

3-CNF No.5: [n=3 k=12] Assignments: [ A1=T A2=T A3=T ] (6 ms)

3-CNF No.6: [n=3 k=12] Assignments: [ A1=F A2=T A3=T ] (87 ms)

3-CNF No.7: [n=3 k=16] No 16-clique; no solution

3-CNF No.8: [n=3 k=16] Assignments: [ A1=T A2=T A3=T ] (2092 ms)