HW8

1. 34.4-7

2-CNF possibilities (x1 v x2) (-x1 v –x2) (-x1 v x2) (x1 v –x2)

Truth table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| X1 | X2 | X1 v x2 | (-x1 v –x2) | (-x1 v x2) | (x1 v –x2) |
| 1 | 1 | 1 | 0 | 1 | 1 |
| 1 | 0 | 1 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 | 1 | 0 |
| 0 | 0 | 0 | 1 | 1 | 1 |

2-CNF is in P since 2-CNF is always satisfiable one variable will always imply the other variable is the same after negation. If x1 is one then x2 will also be one, if x1 is zero then –x2 will be 1.

The book was convoluted so I used the link below as a reference/source.

<http://stackoverflow.com/questions/8467676/how-is-2-cnf-sat-is-in-p-while-3-cnf-sat-is-in-npc>

5. 34-3a

So long as the graph is a bipartite graph then the following pseudo-code should work:

1. Pick a vertex, color that vertex color 1.
2. Color all vertices adjacent to the picked vertex color 2.
3. Go to each adjacent vertex that was colored color 2.
   1. Color all adjacent vertices color 1 – exclude visited vertices.
4. Go to each adjacent vertex that was colored color 1.
   1. Color all adjacent vertices color 2 – exclude visited vertices.
5. Repeat step 3-4 until all vertices are colored.

The follow image was helpful for this problem:

<http://mathworld.wolfram.com/images/eps-gif/BipartiteGraph_1000.gif>