Thrashing

A demo

MyNeq - WordPad

File Edit View Insert Format Help

```
//

public class MyNeq extends AbstractBinIntSConstraint {

    public MyNeq(IntDomainVar x,IntDomainVar y){
      super(x,y);
    }

    public boolean isSatisfied(){
      // System.out.println("isSatisfied()");
      return true;
    }

    public void awake() throws ContradictionException {
      // System.out.println("awake()");
    }

    public void propagate() throws ContradictionException {
      // System.out.println("propagate()");
    }

    public void awakeOnInf(int idx) throws ContradictionException {
    // System.out.println("awakeOnInf(" + idx + ")");
    }

    public void awakeOnSup(int idx) throws ContradictionException {
     // System.out.println("awakeOnSup(" + idx + ")");
    }

    public void awakeOnRem(int idx,int x) throws ContradictionException {
      // System.out.println("awakeOnRem(" + idx + "," + x + ")");
    }

    public void awakeOnInst(int idx) throws ContradictionException {
      if (v0.isInstantiated() && v1.isInstantiated() && v0.getVal() == v1.getVal())
          throw new ContradictionException(v0,-99);
    }
}
```

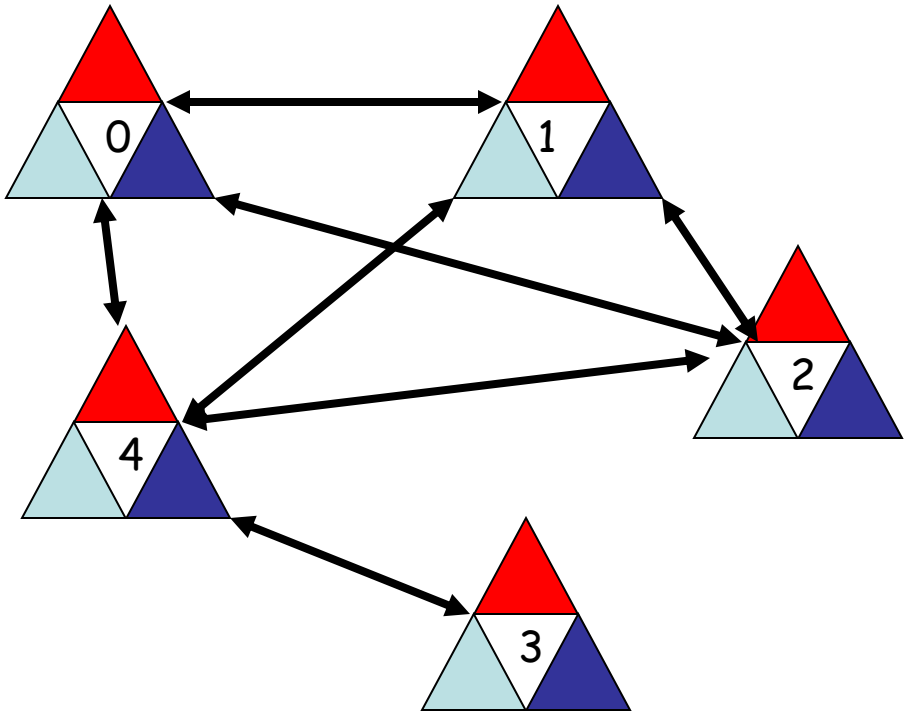
A backward checking constraint

The MyNeq constraints x ≠ y
Constraint is checked only when x and y are instantiated

For Help, press F1

start | thrashing | Command Prompt | 2 Microsoft Of... | MyNeq - WordPad | 12:20

Can you solve this ?



emacs@BIAK

File Edit Options Buffers Tools Java Help

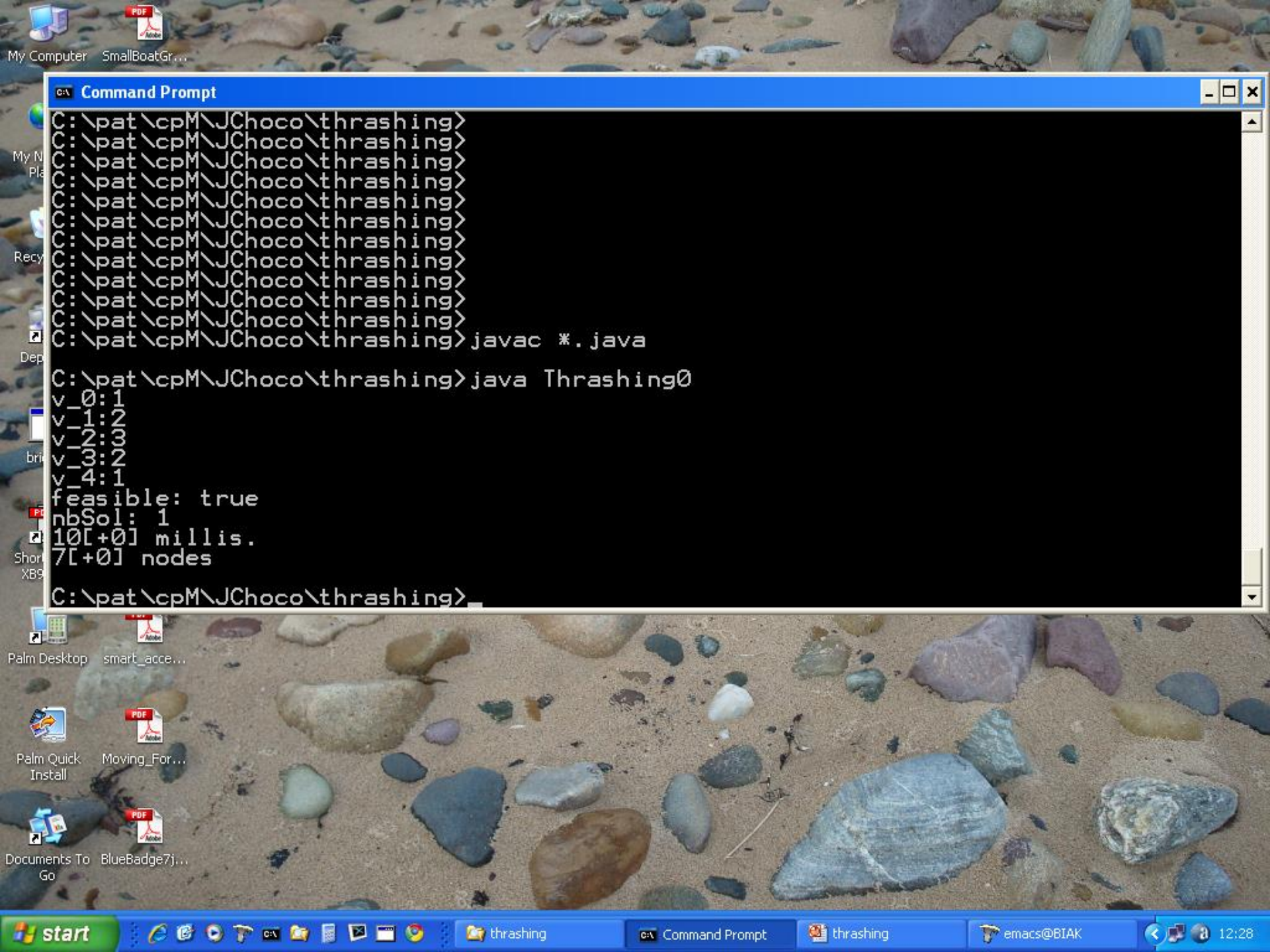
```
import choco.kernel.solver.ContradictionException;
import choco.kernel.model.variables.integer.IntegerVariable;
import choco.cp.solver.search.integer.varselector.StaticVarOrder;

//
//

public class Thrashing0 {

    public static void main(String[] args) throws ContradictionException {

        Model m = new CPModel();
        int n = 5;
        IntegerVariable [] v = makeIntVarArray("v",n,1,3);

        m.addVariables(v);                                    // (1)

        Solver s = new CPSolver();
        s.read(m);
        s.post(new MyNeq(s.getVar(v[0]),s.getVar(v[1])));     // (2)
        s.post(new MyNeq(s.getVar(v[0]),s.getVar(v[2])));
        s.post(new MyNeq(s.getVar(v[1]),s.getVar(v[2])));
        s.post(new MyNeq(s.getVar(v[1]),s.getVar(v[4])));
        s.post(new MyNeq(s.getVar(v[2]),s.getVar(v[4])));
        s.post(new MyNeq(s.getVar(v[3]),s.getVar(v[4])));

        s.setVarIntSelector(new StaticVarOrder(s.getVar(v)));

        s.solve();

        for (int i=0;i<n;i++)
            System.out.println(s.getVar(v[i]));

        System.out.println("feasible: " + s.isFeasible());
        System.out.println("nbSol: " + s.getNbSolutions());
        s.printRuntimeSatistics();
    }
}
//
//
//
```

--\-- Thrashing0.java (Java Abbrev)--L20--13%----------

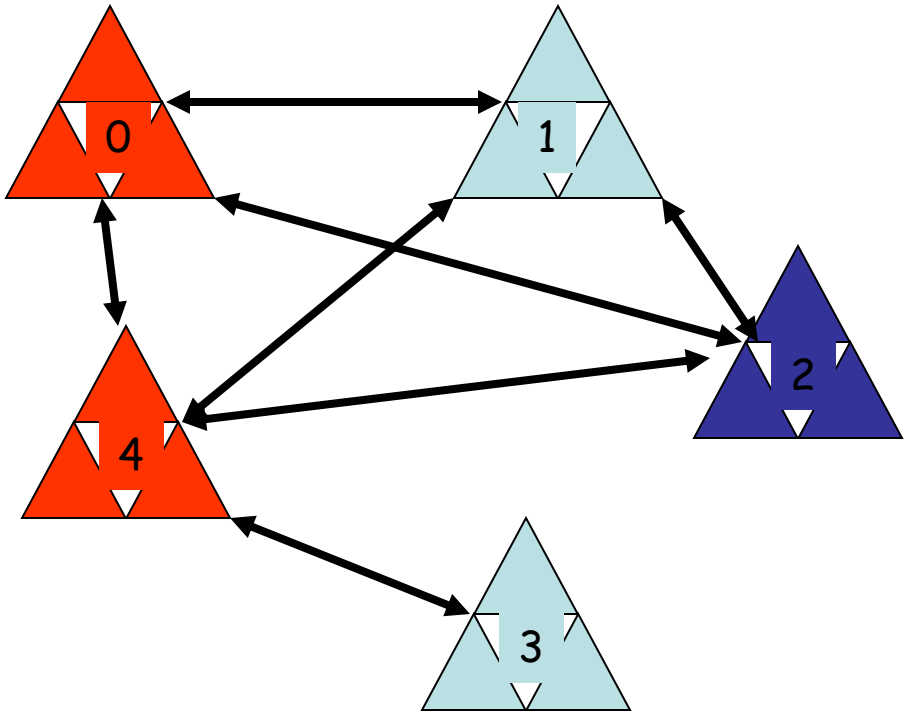
Static variable order
Index order

Red = 1
Blue = 2
Green = 3

start | thrashing | Command Prompt | thrashing | emacs@BIAK | 12:27

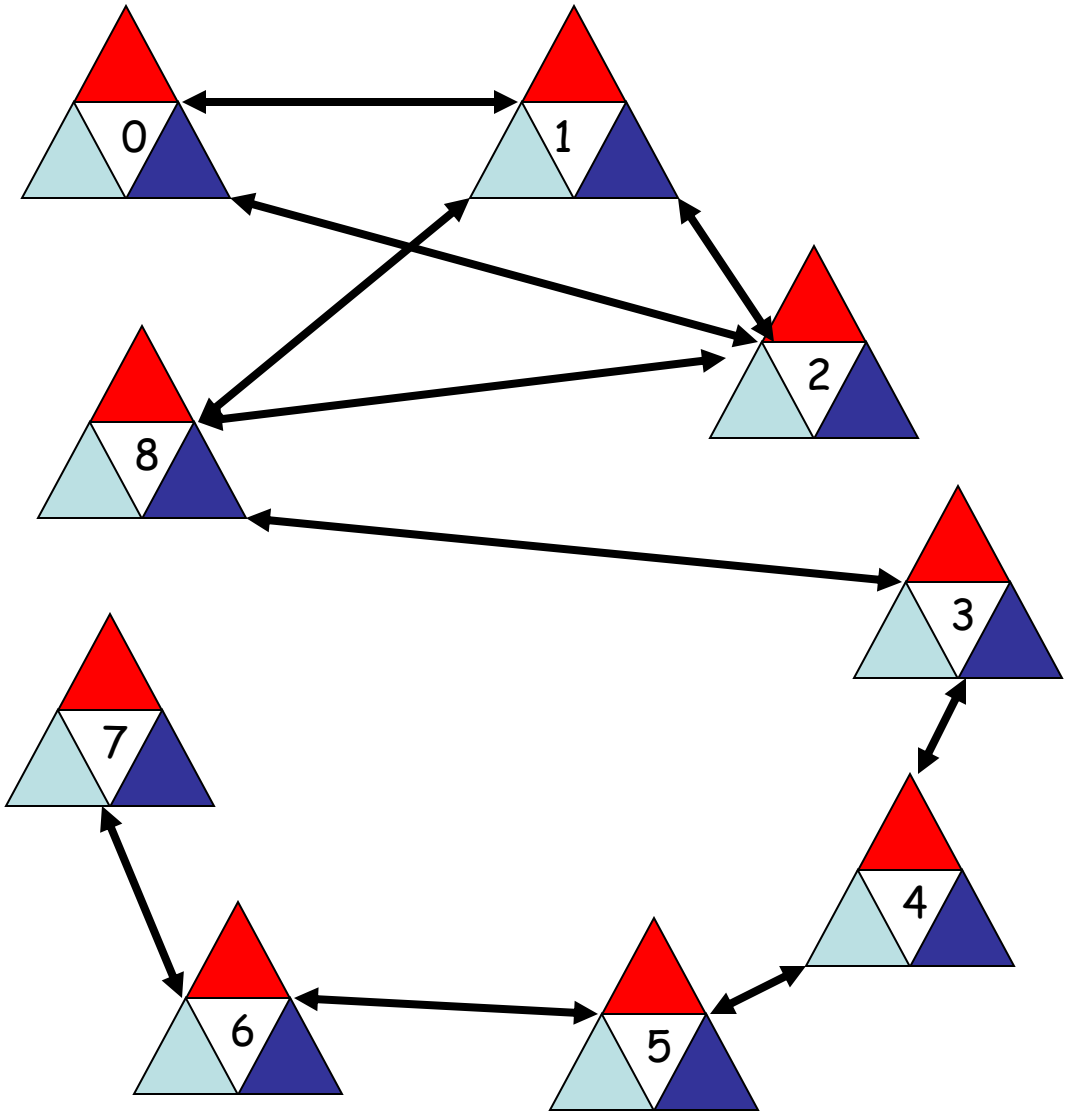

My Computer
SmallBoatGr...
Command Prompt
C:\pat\cpM\JChoco\thrashing>
C:\pat\cpM\JChoco\thrashing>
C:\pat\cpM\JChoco\thrashing>
C:\pat\cpM\JChoco\thrashing>
C:\pat\cpM\JChoco\thrashing>
C:\pat\cpM\JChoco\thrashing>
C:\pat\cpM\JChoco\thrashing>
C:\pat\cpM\JChoco\thrashing>
C:\pat\cpM\JChoco\thrashing>
C:\pat\cpM\JChoco\thrashing>
C:\pat\cpM\JChoco\thrashing>javac *.java
C:\pat\cpM\JChoco\thrashing>java Thrashing0
v_0:1
v_1:2
v_2:3
v_3:2
v_4:1
feasible: true
nbSol: 1
10[+0] millis.
7[+0] nodes
C:\pat\cpM\JChoco\thrashing>
Palm Desktop
smart_acce...
Palm Quick Install
Moving_For...
Documents To Go
BlueBadge7j...
start
thrashing
Command Prompt
thrashing
emacs@BIAK
12:28

Solution

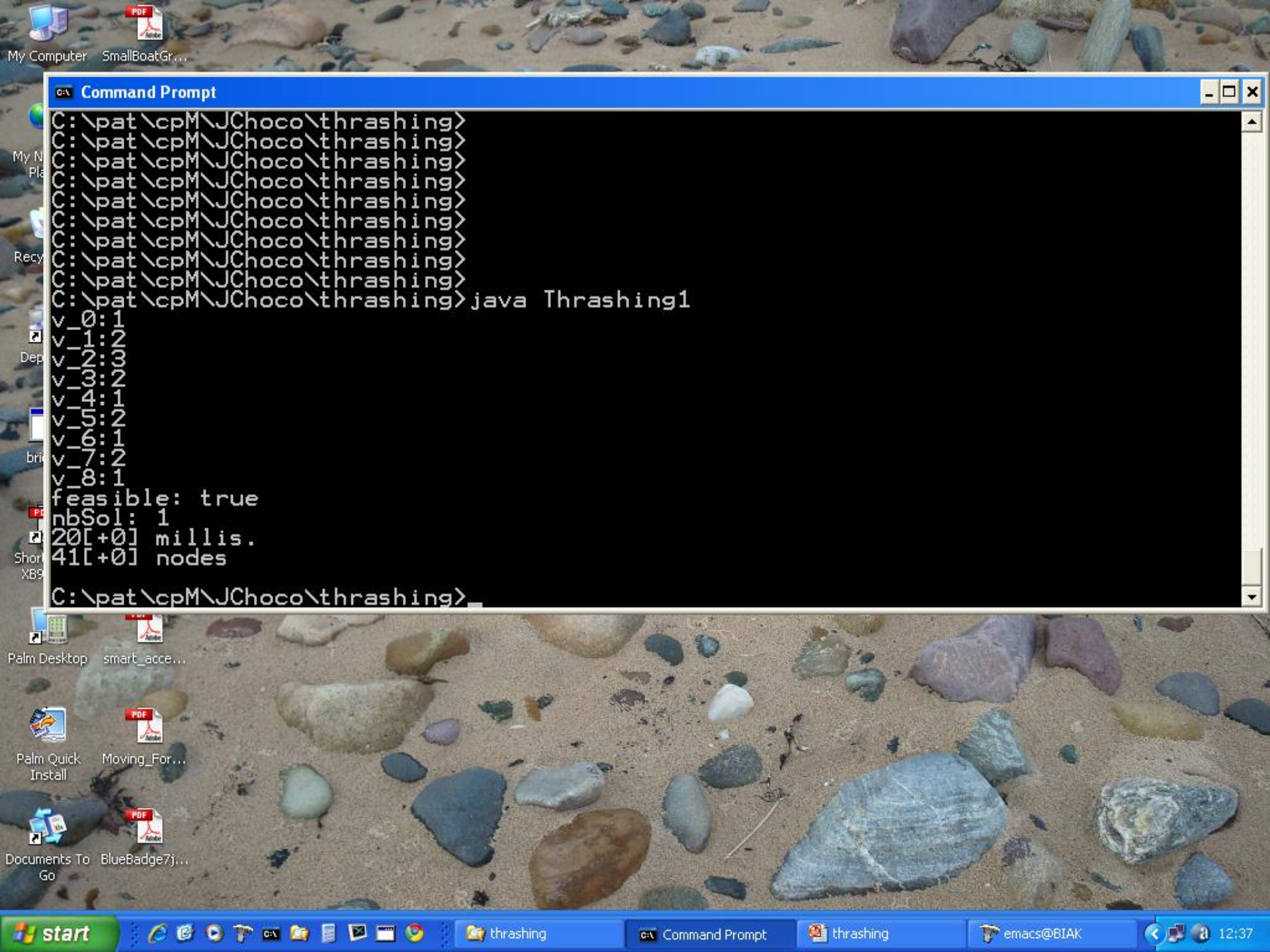


Thrashing0

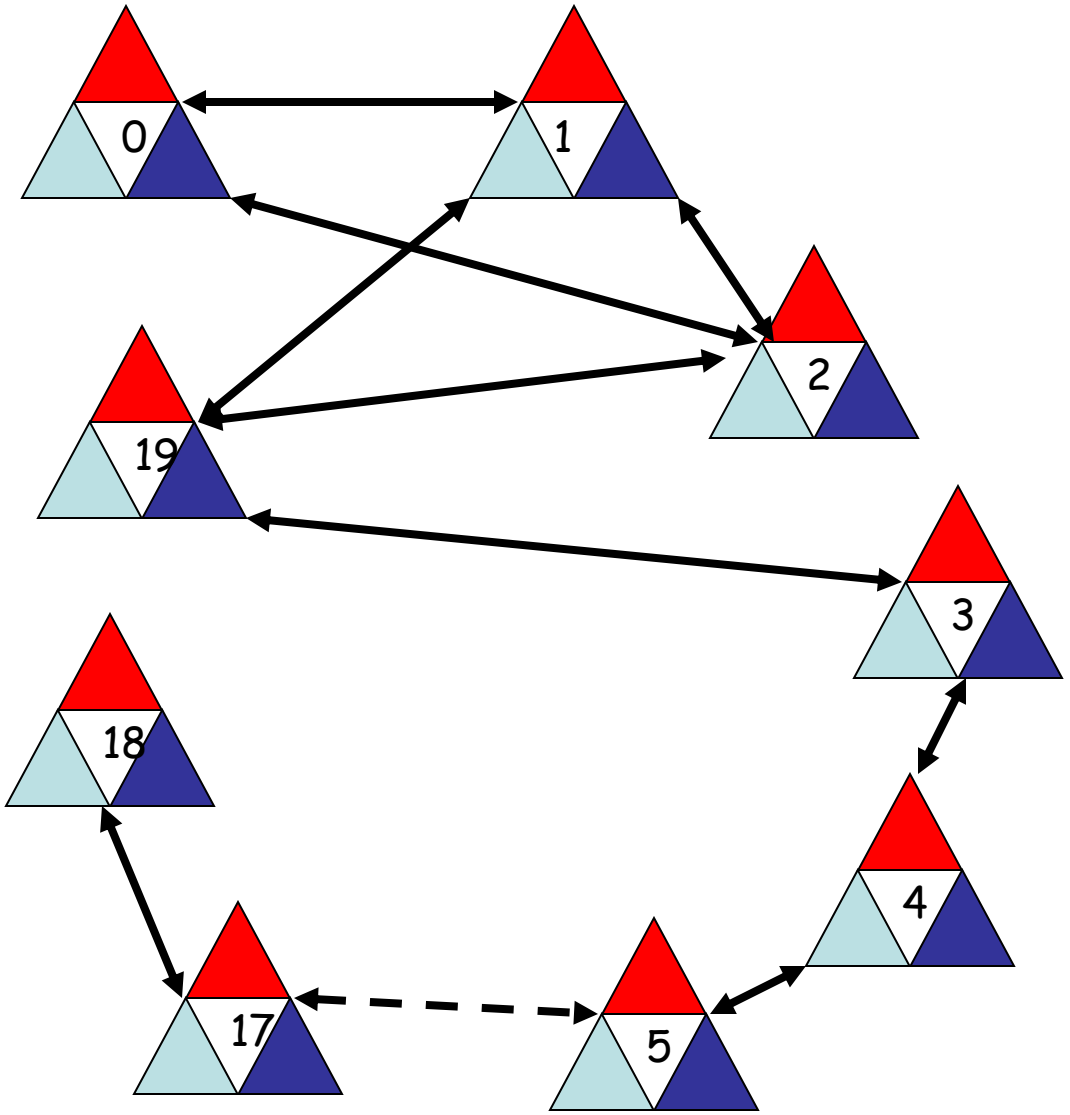
Thrashing1



```
//
//

public class Thrashing1 {

    public static void main(String[] args) throws ContradictionException {

        Model m = new CPModel();
        int n = 9;
        IntegerVariable [] v = makeIntVarArray("v",n,1,3);

        m.addVariables(v);                                    // (1)

        Solver s = new CPSolver();
        s.read(m);
        s.post(new MyNeq(s.getVar(v[0]),s.getVar(v[1])));     // (2)
        s.post(new MyNeq(s.getVar(v[0]),s.getVar(v[2])));
        s.post(new MyNeq(s.getVar(v[1]),s.getVar(v[2])));
        s.post(new MyNeq(s.getVar(v[1]),s.getVar(v[8])));
        s.post(new MyNeq(s.getVar(v[2]),s.getVar(v[8])));
        s.post(new MyNeq(s.getVar(v[3]),s.getVar(v[8])));
        s.post(new MyNeq(s.getVar(v[3]),s.getVar(v[4])));
        s.post(new MyNeq(s.getVar(v[4]),s.getVar(v[5])));
        s.post(new MyNeq(s.getVar(v[5]),s.getVar(v[6])));
        s.post(new MyNeq(s.getVar(v[6]),s.getVar(v[7])));

        s.setVarIntSelector(new StaticVarOrder(s.getVar(v)));

        s.solve();

        for (int i=0;i<n;i++)
            System.out.println(s.getVar(v[i]));

        System.out.println("feasible: " + s.isFeasible());
        System.out.println("nbSol: " + s.getNbSolutions());
        s.printRuntimeSatistics();
    }
}
//
//
```


My Computer
SmallBoatGr...
Command Prompt
C:\pat\cpM\JChoco\thrashing>
C:\pat\cpM\JChoco\thrashing>
C:\pat\cpM\JChoco\thrashing>
C:\pat\cpM\JChoco\thrashing>
C:\pat\cpM\JChoco\thrashing>
C:\pat\cpM\JChoco\thrashing>
C:\pat\cpM\JChoco\thrashing>
C:\pat\cpM\JChoco\thrashing>
C:\pat\cpM\JChoco\thrashing>
C:\pat\cpM\JChoco\thrashing>java Thrashing1
v_0:1
v_1:2
v_2:3
v_3:2
v_4:1
v_5:2
v_6:1
v_7:2
v_8:1
feasible: true
nbSol: 1
20[+0] millis.
41[+0] nodes
C:\pat\cpM\JChoco\thrashing>
Palm Desktop
smart_acce...
Palm Quick Install
Moving_For...
Documents To Go
BlueBadge7j...
start
thrashing
Command Prompt
thrashing
emacs@BIAK
12:37

Thrashing2



emacs@BIAK

File Edit Options Buffers Tools Java Help

```
//
//

public class Thrashing2 {

    public static void main(String[] args) throws ContradictionException {

        Model m = new CPModel();
        int n = 20;
        IntegerVariable [] v = makeIntVarArray("v",n,1,3);

        m.addVariables(v);                                    // (1)

        Solver s = new CPSolver();
        s.read(m);
        s.post(new MyNeq(s.getVar(v[0]),s.getVar(v[1])));     // (2)
        s.post(new MyNeq(s.getVar(v[0]),s.getVar(v[2])));
        s.post(new MyNeq(s.getVar(v[1]),s.getVar(v[2])));
        s.post(new MyNeq(s.getVar(v[1]),s.getVar(v[19])));
        s.post(new MyNeq(s.getVar(v[2]),s.getVar(v[19])));
        s.post(new MyNeq(s.getVar(v[3]),s.getVar(v[19])));
        s.post(new MyNeq(s.getVar(v[3]),s.getVar(v[4])));
        for (int i=4;i<n-1;i++)
            s.post(new MyNeq(s.getVar(v[i]),s.getVar(v[i+1])));

        s.setVarIntSelector(new StaticVarOrder(s.getVar(v)));

        s.solve();

        for (int i=0;i<n;i++)
            System.out.println(s.getVar(v[i]));

        System.out.println("feasible: " + s.isFeasible());
        System.out.println("nbSol: " + s.getNbSolutions());
        s.printRuntimeSatistics();
    }
}
//
//
//
```

20 variables

--\-- **Thrashing2.java** (Java Abbrev)--L9--25%----------

start | thrashing | Command Prompt | thrashing | emacs@BIAK | 12:39

Run it
Feel the pain

Two possible changes of interest

1. Use a different instantiation order
   Edit out static order
2. Use neq rather than MyNeq
   Check forwards!