Computability

CPS 001

Mason F. Matthews

October 19th and 21st, 2005

Write Your Names

(or just exercise your curiosity)

What is Computer Science?

What can a computer do?

What can a computer not do?

Types of Problems

- Tractable
  - Problems that **can** be solved by a computer in a "reasonable" amount of time.

Types of Problems

- Tractable
  - Problems that **can** be solved by a computer in a "reasonable" amount of time.
- Intractable
  - Problems that **can't** be solved by a computer in a "reasonable" amount of time,
  - But **can** be solved eventually.

Types of Problems

- Tractable
  - Problems that **can** be solved by a computer in a "reasonable" amount of time.
- Intractable
  - Problems that **can't** be solved by a computer in a "reasonable" amount of time,
  - But **can** be solved eventually.
- Non-computable
  - Problems that can **never** be solved by a computer.

Questions?

Questions?

- What is a "problem?"

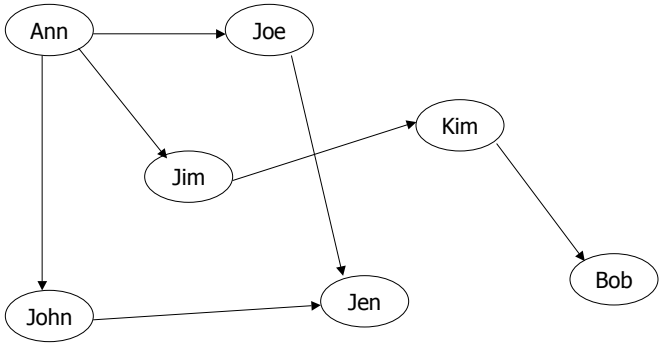
Questions?

- What is a "problem?"
- What if I do something dumb, and it takes forever?

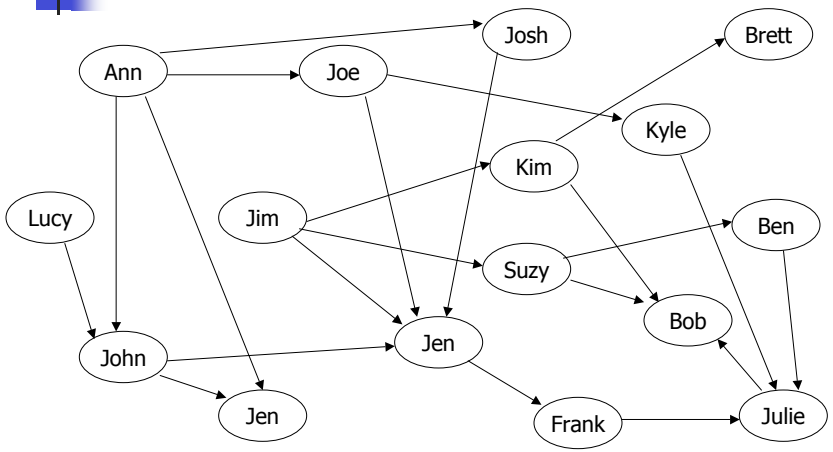
Questions?

- What is a "problem?"
- What if I do something dumb, and it takes forever?
- What does "reasonable" mean?

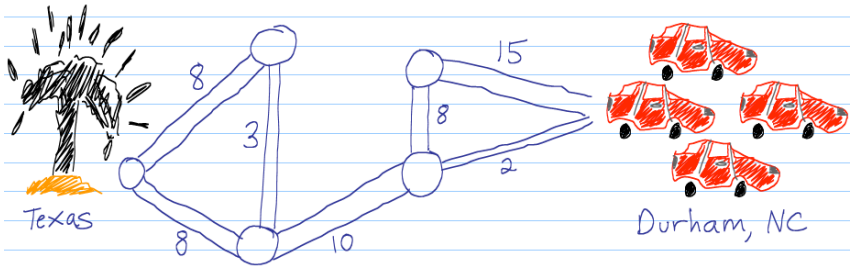
Is there a path from Ann to Bob?



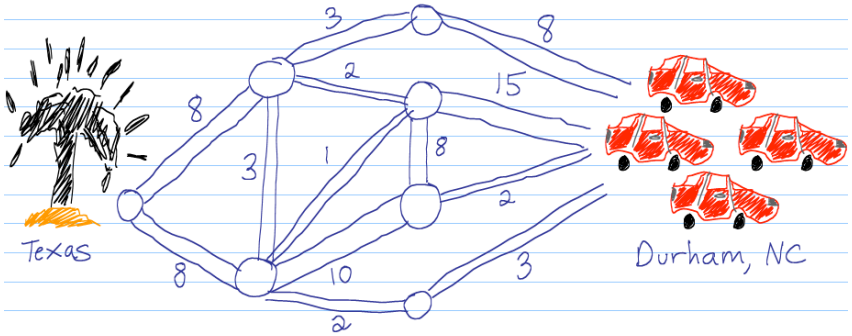
Is there a path from Ann to Bob?



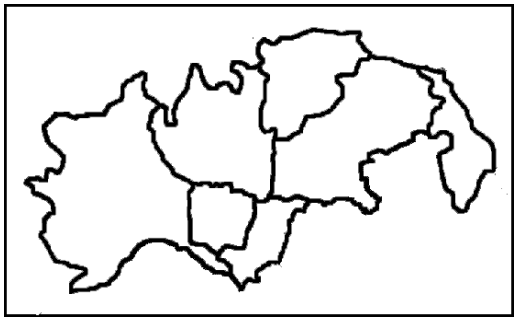
How much oil can flow?



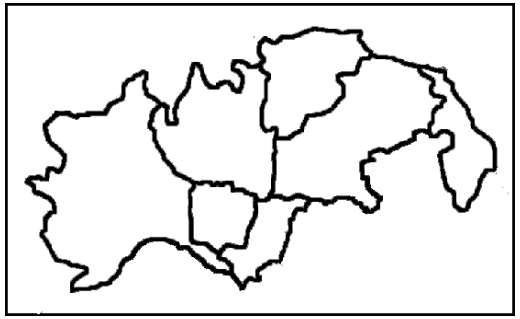
How much oil can flow?



Can you color this map with 4 colors?

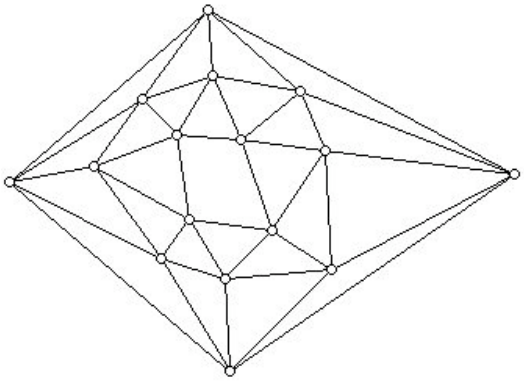


Can you color this map with 3 colors?

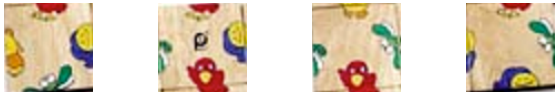


Can you color this map with 3 colors?

Can you color this map with 3 colors?



Can you solve this puzzle?



Can you solve this puzzle?

Can you solve this puzzle?

- Does "Intractable" mean impossible?

What is the hardest problem that you can think of?

How can we deal with these problems?

How can we deal with these problems?

- Random Numbers
  - Can "expect" to solve some in reasonable time

How can we deal with these problems?

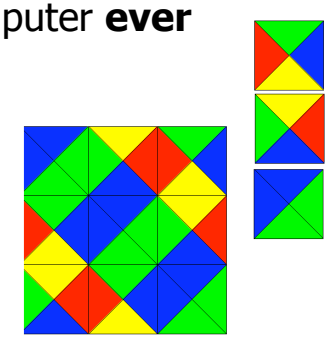
- Random Numbers
  - Can "expect" to solve some in reasonable time
- Approximation
  - Can guarantee that we're "close" to the right answer

How can we deal with these problems?

- Random Numbers
  - Can "expect" to solve some in reasonable time
- Approximation
  - Can guarantee that we're "close" to the right answer
- Parallel Computers?

Non-Computable Problems

- Problems that **cannot** be solved by a computer **ever**



If you start this program, will it ever stop running?

```
while(1==1) {
  System.out.println("Still Running.");
}
return;
```

If you start this program, will it ever stop running?

```
int i = 1;
while(i>0) {
  System.out.println("Still Running.");
  i++;
}
return;
```

If you start this program, will it ever stop running?

```
public class Client {
        public static void main(String[] args) {
            if(args.length != 2) {
                System.out.println("Usage: vbj Client <carrier-name> <aircraft-name>\n");
                return;
            }
    String carrierName = args[0];
            String aircraftName = args[1];
            org.omg.CORBA.Object carrier = null;
            org.omg.CORBA.Object aircraft = null;
    org.omg.CORBA.ORB orb = null;
            try {
                orb = org.omg.CORBA.ORB.init(args, null);
            }
            catch (org.omg.CORBA.SystemException se) {
                System.err.println("ORB init failure " + se);
                System.exit(1);
            }
    { // scope
                try {
                    carrier = orb.bind("IDL:Ship/AircraftCarrier:1.0", carrierName, null, null);
                }
                catch (org.omg.CORBA.SystemException se) {
                    System.err.println("ORB init failure " + se);
                    System.exit(1);
                }
                org.omg.CORBA.Request request = carrier._request("launch");
                request.add_in_arg().insert_string(aircraftName);

      request.set_return_type(orb.get_primitive_tc(org.omg.CORBA.TCKind.tk_objref));
                request.invoke();
                aircraft = request.result().value().extract_Object();
            }
    { // scope
                org.omg.CORBA.Request request = aircraft._request("codeNumber");

      request.set_return_type(orb.get_primitive_tc(org.omg.CORBA.TCKind.tk_string));
                request.invoke();
                String designation = request.result().value().extract_string();
                System.out.println ("Aircraft " + designation + " is coming your way");
            }
    { // scope
                org.omg.CORBA.Request request = aircraft._request("attitude");
      int altitude = 10000;
                org.omg.CORBA.Any ioAltitude = request.add_inout_arg();
                ioAltitude.insert_long(altitude);

      try {
                    carrier = orb.bind("IDL:Ship/AircraftCarrier:1.0", carrierName, null, null);
                }
                catch (org.omg.CORBA.SystemException se) {
                    System.err.println("ORB init failure " + se);
                    System.exit(1);
                }
                org.omg.CORBA.Request request = carrier._request("launch");
                request.add_in_arg().insert_string(aircraftName);

      request.set_return_type(orb.get_primitive_tc(org.omg.CORBA.TCKind.tk_objref));
                request.invoke();
                aircraft = request.result().value().extract_Object();
            }
    { // scope
                org.omg.CORBA.Request request = aircraft._request("codeNumber");

      request.set_return_type(orb.get_primitive_tc(org.omg.CORBA.TCKind.tk_string));
                request.invoke();
                String designation = request.result().value().extract_string();
                System.out.println ("Aircraft " + designation + " is coming your way");
            }
    { // scope
                org.omg.CORBA.Request request = aircraft._request("attitude");
      int altitude = 10000;
                org.omg.CORBA.Any ioAltitude = request.add_inout_arg();
                ioAltitude.insert_long(altitude);
      String direction = "headup";
                request.add_in_arg().insert_string(direction);
                request.invoke();
      altitude = ioAltitude.extract_long();
      System.out.println ("Aircraft is heading up to " + altitude + " Feet.");
            }
        }
    }
```

Proof of impossibility!

What else can't we know about our programs?

- Do two programs do the same thing?
- Do programs have any bugs?
- Do programs do what they're supposed to do?

What else can't we know about our programs?

- Do two programs do the same thing?
- Do programs have any bugs?
- Do programs do what they're supposed to do?