

Today's topics

Digital Intellectual Property Issues

Programming

- Recursion
- Invariants

Reading

Intellectual Property Readings

Intellectual property for CS students

Software patents: One-click and Air Travel

Open Source definitions

Microsoft Corp., "Some Questions Every Business Should Ask About the GNU General Public License (GPL)", 2001

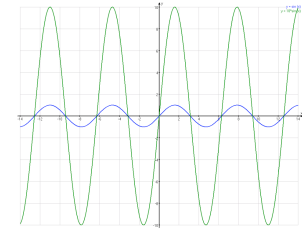
Brookshear, Chapter 6

CPS 001

8.1

What is digital?

- What's the difference between
 - > Rolex and Timex?
 - > VCR tape and DVD?
- How is ripping to a mp3 different from recording to a tape?
 - > Reproduction: immediate and future
 - > Distribution
 - > Modification
- Why do digital media present new challenges from analog media?
 - > Is copyright infringement new?



CPS 001

8.2

Copyright

- US Constitution (Article I, Section 8, Clause 8):
"To promote the Progress of Science and useful Arts"
 - > What can you copyright?
 - Fixed, tangible medium of expression with a modicum of originality
 - > How do you copyright?
 - Don't need anything. Registration necessary for copyright infringement suits
 - > Authors given limited monopoly so they will disclose to public
 - > Concessions
 1. Fair use
 2. First sale
 3. Limited Time
 - > Evolving Bargain: Copyright holder may profit from works and public has access and can build upon them
 - > What would happen if information could only be shared if the owner provided permission?

CPS 001

8.3

Fair use

- Use copyrighted works without permission if the use does not unduly interfere with the copyright owner's market for a work
- Include personal, noncommercial uses
- 4 prong test
 1. Purpose and character of use (commercial vs. non-profit or educational)
 2. Nature of copyrighted work
 3. Amount and substantiality of the portion used
 4. Effect of the copying upon market
- Example: using a VCR to time-shift a broadcast program
- Reverse engineering
 - > OK when extracting unprotected elements
 - > Connectix Virtual PlayStation

CPS 001

8.4

Digital rights management

- Idea: copying is hard to control, so make the copying process itself difficult
 - Restrict the use of digital files in order to protect interest of copyright holders
 - Control file access
 - Implemented in operating system, program software, or in the actual hardware of a device
- Digital watermarking
 - Make information so that unauthorized copying can be detected
- Serial Copy Management System (Audio Home Recording Act 92)
- Dystopian and utopian results?
- Privacy issues?

CPS 001

8.5

Test Case

Artist Def Jeff creates a new song that includes a four-bar percussion sample from a Rolling Stones song. He uses the sample without receiving permission from the copyright holder, His DJ overlays the track containing the sample with 40 tracks of original music and puts the song on his latest CD.

- Were Def Jeff's actions within the limits of fair use?
- Would he lose a lawsuit?
- Were his actions ethical?

CPS 001

8.6

Important papers

- National Information Infrastructure White Paper 1995
 1. Copyright owners given exclusive rights over "transmission" of information not just copying
 2. Eliminate first-sale doctrine for digital works
 3. Criminalize tampering with copyright protection or identification mechanisms
 - Controversial and bills to implement recommendations were not passed, until...
- World Intellectual Property Organization Treaty 1996
- Digital Millennium Copyright Act 1998
 - Encourages use of technological protections to facilitate a pay-per-view/use system
 - Copyright used to regulate multiplication and distribution of works but how about consumption?
 - Civil and criminal penalties for circumventing copyright protection systems

CPS 001

8.7

Copyrights

- Copyright Term Extension Act 1998
 - Free Mickey Mouse! (challenged in Supreme Court 2003)
 - Retroactive copyright extension of 20 years
 - Breyer: "effect ... is to make the copyright term not limited, but virtually perpetual"
 - Over the last 40 years, Congress has lengthened copyright durations 11 times
 - Copyright term length
 - 95 years for corporations
 - 70 years after death for individuals
- Other forms of exclusive rights in information
 - Patents: inventions that others cannot use
 - Trademark: differentiates between different sources of products
 - Trade secret: pseudo-property right to penalize those who disclose information to unauthorized persons

CPS 001

8.8

Questions

- Is copyright infringement stealing?
- What are the differences between writing code and writing books in terms of licensing?
- Discuss the legality of peer-to-peer sharing with respect to the four prongs of determining fair use
- **Eben Moglen:**
 - > If you could feed everyone by pressing a button to create lambchops (for free), is there a moral reason to have starving people?
 - > If everything has zero marginal cost and can be given to everyone everywhere why is it ever moral to exclude anyone from anything?

CPS 001

8.9






Consequences

- **Scientific research**
 - > Secure Digital Music Initiative & Prof. Edward Felton
 - > Adobe & Dmitry Sklyarov
- **Fair Use**
 - > Copy-protected CDs
 - > DeCSS and DVD Copy Plus
- **Innovation and competition**
 - > Sony vs. Connectix and "Mod Chip" makers
 - > Apple & Other World Computing

CPS 001

8.10

Patents

- **Why patents are powerful?**
 - > Right to exclude others from "practicing the invention"
- **Currently operating under Patent Act of 1975**
 - > 20 year term
- **Patent and Trademark Office looks at 4 criteria**
 -  Is proposed invention patentable?
 -  Utility
 -  Novelty
 -  Non-obviousness
-  **Software patents**
 - > Only recently have patents been granted for software or business methods
 - > Controversial patent: Amazon.com's One-Click

CPS 001

8.11

Types of software

- **Software Licenses**
 - > Public domain
 - > Free
 - > Open Source
 - > Copylefted
 - > Semi-free
 - > Commercial
- **Proprietary (closed) software**
 - > Freeware
 - > Shareware
 - > Adware
 - > Spyware
- **Commercial**
 - Academic licenses
- **Specific licenses**
 - > GNU Public License
 - > Mozilla Public License
 - > Lots more...

Say you buy software using (steep) student discounts that are available at the Duke Computer Store?

Can you use it when you leave Duke and are no longer a student?

What do you buy when you purchase software?

CPS 001

8.12

Open source

- **Commercial software license schemes**
 - > Microsoft's Embrace and Extend
 - > What's a EULA?
- **Rights**
 - > Make copies of the program and distribute them
 - > Access to the software's source code
 - > Make improvements to the program
- **Results**
 - > All contributors at same relative level
 - > Lots of competition in distribution or support
 - > *Why does it work?*
- **Free Software Foundation formed in 1984**
 - > GNU General Public License (Copyleft)
 - > Seminal work produced (emacs, gnu compiler)
 - > Spawned different licenses like the [Open Source Definition](#)

CPS 001

8.13

Sources of material

- **Organizations**
 - > [The Electronic Frontier Foundation](#)
 - > [Center for Democracy and Technology](#)
- **Media and discussion**
 - > [Wired Magazine](#)
 - > [Slashdot](#)
- **Databases of information and laws**
 - > [Lexis/Nexis](#)
 - > [Thomas](#)
- **Social issues in Computer Science**
 - > [Computer Professionals for Social Responsibility](#)

CPS 001

8.14

Solving Problems Recursively

- **Recursion is an indispensable tool in a programmer's toolkit**
 - > Allows many complex problems to be solved simply
 - > Elegance and understanding in code often leads to better programs: easier to modify, extend, verify
 - > Sometimes recursion isn't appropriate, when it's bad it can be very bad---every tool requires knowledge and experience in how to use it
- **The basic idea is to get help solving a problem from coworkers (clones) who work and act like you do**
 - > Ask clone to solve a simpler but similar problem
 - > Use clone's result to put together your answer
- **Need both concepts: call on the clone and use the result**

CPS 001

8.15

Fundamentals of Recursion

- **Base case (aka exit case)**
 - > Simple case that can be solved with no further computation
 - > Does not make a recursive call
- **Reduction step (aka Inductive hypothesis)**
 - > Reduce the problem to another *smaller* one of the same structure
 - > Make a recursive call, with some parameter or other measure that decreases or moves towards the base case
 - Ensure that sequence of calls eventually reaches the base case
 - "Measure" can be tricky, but usually it's straightforward
- **The Leap of Faith!**
 - > If it works for the reduction step is correct and there is proper handling of the base case, the recursion is correct.
- **What row are you in?**

CPS 001

8.16

Classic examples of recursion

- For some reason, computer science uses these examples:
 - > Factorial: we can use a loop or recursion, is this an issue?
 - > Fibonacci numbers: 1, 1, 2, 3, 5, 8, 13, 21, ...
 - $F(n) = F(n-1) + F(n-2)$, why isn't this enough? What's needed?
 - Classic example of bad recursion, to compute $F(6)$, the sixth Fibonacci number, we must compute $F(5)$ and $F(4)$. What do we do to compute $F(5)$? Why is this a problem?
 - > Towers of Hanoi
 - N disks on one of three pegs, transfer all disks to another peg, never put a disk on a smaller one, only on larger
 - Every solution takes "forever" when N , number of disks, is large
 - > Reversing strings
 - Append first character after the rest is reversed

CPS 001

8.17

Exponentiation

- Computing x^n means multiplying n numbers (or does it?)
 - > What's the easiest value of n to compute x^n ?
 - > If you want to multiply only once, what can you ask a clone?

```
double Power(double x, int n)
// post: returns x^n
{
    if (n == 0)
    {
        return 1.0;
    }
    return x * Power(x, n-1);
}
```

- What about an iterative version?

CPS 001

8.18

Faster exponentiation

- How many recursive calls are made to compute 2^{1024} ?
 - > How many multiplies on each call? Is this better?

```
double Power(double x, int n)
// post: returns x^n
{
    if (n == 0)
    {
        return 1.0;
    }
    double semi = Power(x, n/2);
    if (n % 2 == 0)
    {
        return semi*semi;
    }
    return x * semi * semi;
}
```

- What about an iterative version of this function?

CPS 001

8.19

Loop Invariants

- Want to reason about the correctness of a proposed iterative solution
- Loop invariants provide a means to effectively about the correctness of code

```
while !done do
{
    // what is true at every step
    // Update/iterate
    // maintain invariant
}
```

CPS 001

8.20

Bean Can game

- Can contains N black beans and M white beans initially
- Emptied according the following repeated process
 - > Select two beans from the can
 - > If the beans are:
 - *same color*: put a black bean back in the can
 - *different colors*: put a white bean back in the can
 - > Player who chooses the color of the remaining bean wins the game
- Analyze the link between the initial state and the final state
- Identify a property that is preserved as beans are removed from the can
 - > *Invariant* that characterizes the removal process