CENTER FOR TEXTUAL STUDIES AND
DIGITAL HUMANITIES

DIGH 401 - Introduction to Computing

Fall Semester 2013

Week 2

Today's Class

Part 1

- Computing Power, Information & Values

Part 2

- Languages and Programming
- A few basics
- Getting started with languages and programming

Part 1 - Computing Power

- mechanical work with horsepower & watt

- measuring the power of a computing machine

1. How much information it can process?
2. How fast can it process?
3. Efficiency

How much information it can process?

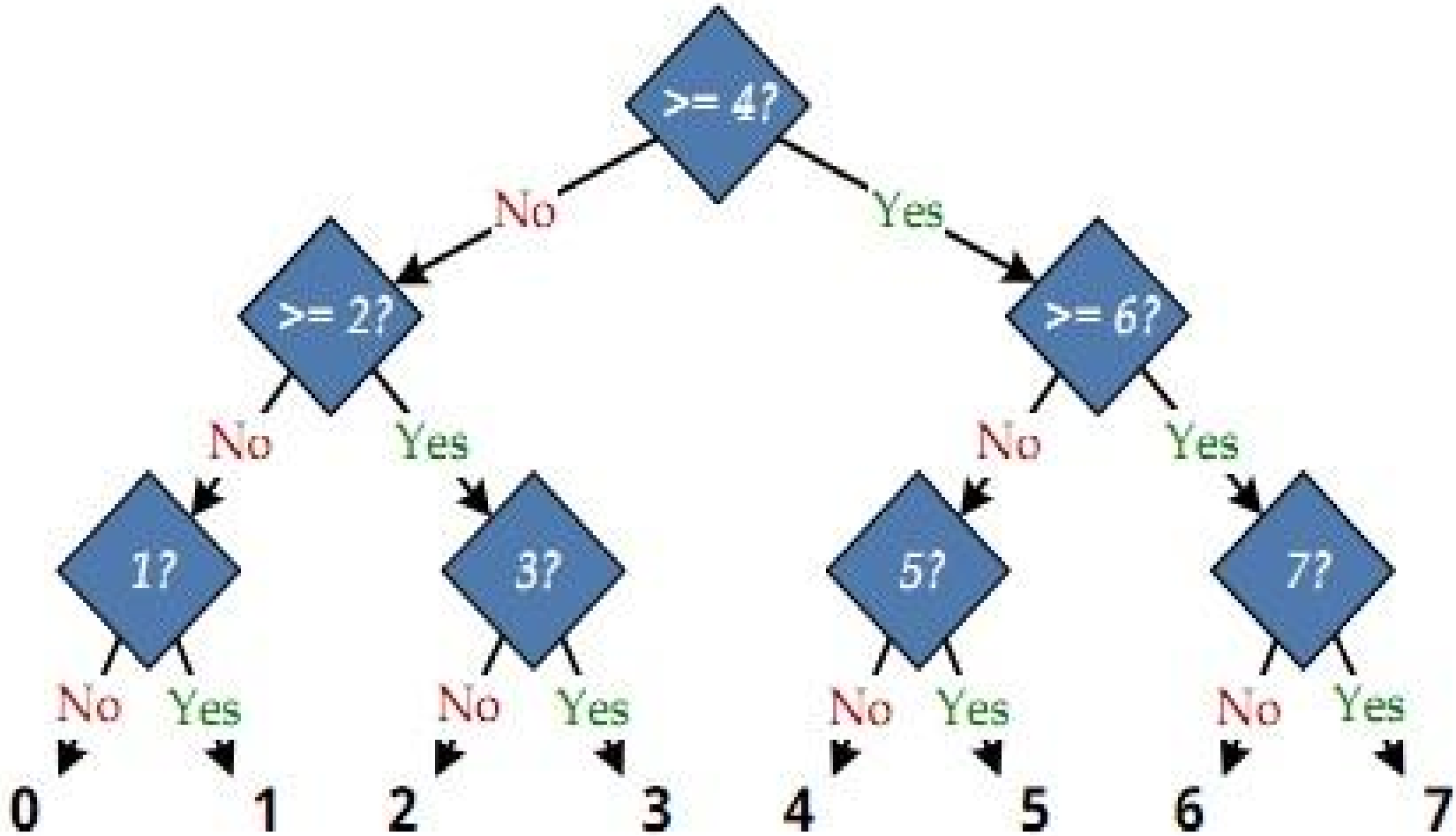
- informally we use information to mean knowledge
- primary unit of information is a 'bit'
- what is a binary question?
- tossing a coin

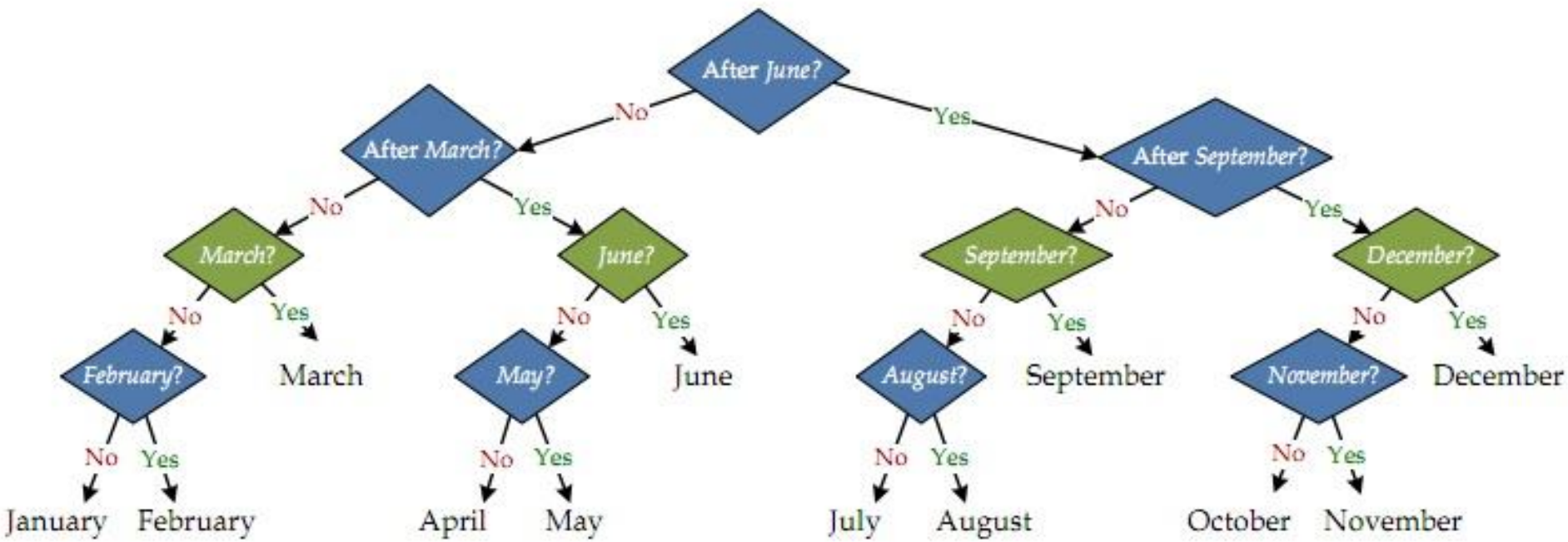
Binary questions

- goal is to identify questions where the “yes” and “no” answers are equally likely

- 2 more bits would allow us to distinguish between 4 possible outcomes

- questions are discrete, so we round up


>= 4?
No
Yes
>= 2?
>= 6?
No
Yes
No
Yes
1?
3?
5?
7?
No
Yes
No
Yes
No
Yes
No
Yes
0
1
2
3
4
5
6
7


After June?
No
Yes
After March?
After September?
No
Yes
No
Yes
March?
June?
September?
December?
No
Yes
No
Yes
No
Yes
No
Yes
February?
March
May?
June
August?
September
November?
December
No
Yes
No
Yes
No
Yes
No
Yes
January
February
April
May
July
August
October
November

How a computer sees text and images

Text

- enumerate all texts alphabetically by length

a, b, c, . . ., z, aa, ab, . . ., az, ba, . . ., zz, aaa, . .

- for example, we could work with 57 different symbols from the English language
    - 26 lower-case
    - 26 upper-case
    - space, comma, period, new line, semi-colon

a	000000
b	000001
c	000010
d	000011
...	...
p	001111
q	010000
...	...
z	011001

A	011010
B	011011
C	011100
...	...
F	011111
G	100000
...	...
Y	110010
Z	110011

space	110100
,	110101
.	110110
newline	110111
;	111000
unused	111001
...	...
unused	111110
unused	111111

- raw editing of a text

How a computer sees text and images

Images

- eg: a black and white picture

-

```
0000011111100000
0000100000010000
0011000000001100
0010000000000100
       ...
0000011111100000
```

Part 2 - Languages and Programming

- Examine various levels of computer programming

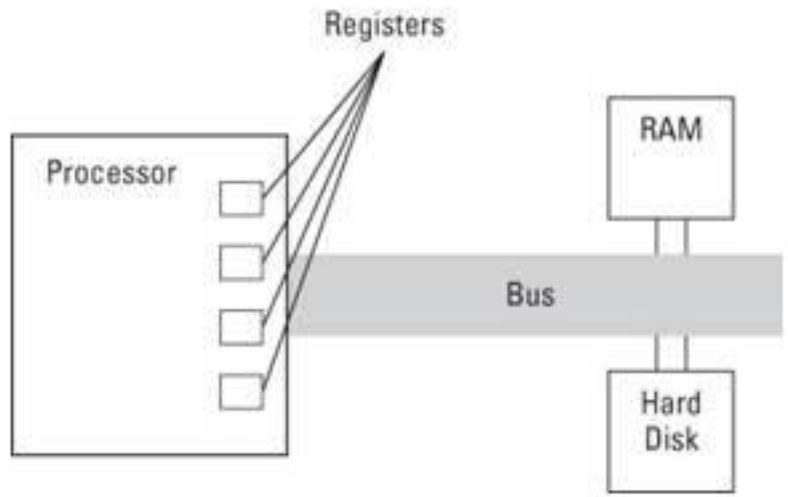
- processors and machine language

1011 0000 0110 0001

- tedious and rarely used

Assembly language

- makes programming easier than machine language
- by itself, a processor is pretty useless for computational work
- communication through a 'bus'
- temporary storage in the 'register'



Assembly language

- example of assembly language

```
sub al, 061h
```

- still overly complicated and not suited to large application development

- even for small programs you still need to manipulate the registers

- add two numbers together with assembly language

High level languages

- aim was to make programming more intuitive

- create a layer of abstraction

- 'Hello World'

Java 'Hello World'

```
class helloworld
{
        public static void main(String args[])
        {
           System.out.println("Hello World!");
        }
}
```

PHP 'Hello World'

```
<?php

echo "Hello World, how are you?";

?>
```

http://dhdev.ctsdh.luc.edu/testing/digh401/week2/

Compilers

- high level language into machine language

- translate using a compiler

- trade-offs from high level to machine language

Getting started with programming

- what do you need to get started?

    - choose an OS
    - choose a preferred editor
    - select your programming language
    - compiler will be relative to the selected language

Getting started with programming

eg: Java programming

- A Java compiler and virtual machine
    - available from www.java.com
- An editor such as Eclipse
    - available from www.eclipse.org

Getting started with programming

eg: PHP programming

- a basic text editor to create and save .php files
    - eg: TextWrangler or BBEdit on OS X...

- a PHP enabled web server, such as Apache 2 with PHP support
    - XAMPP

- a web browser to view the output

Getting started with programming

- describe how programming works

- employ examples from various different languages

- begin practicing and initial programming with a specific language

Getting started with PHP programming

- PHP:Hypertext Preprocessor

- server-side language
  - PHP code is executed on the server

- support for many different databases including MySQL, PostgreSQL...

- open source software
  - free to download and use
  - included with LAMP stacks, XAMPP test packages...
  - http://www.php.net

Getting started with PHP programming

- runs on multiple platforms including Linux, OS X, Windows, Unix...

- compatible with many web servers such as Apache, IIS...

- files can include different content such as
    - text, HTML tags, scripts, styling...

- PHP files are interpreted by a server and returned to the browser as plain HTML

- files normally end in .php

Getting started with PHP programming

```
<?php
?>
```

- can be stand-alone or embedded in HTML, for example

```
<html>
<body>
<?php
echo "Hello World, how are you?";
?>
</body>
</html>
```

- each code line or grouping must end with a semi-colon
    - distinguishes one set of instructions from another