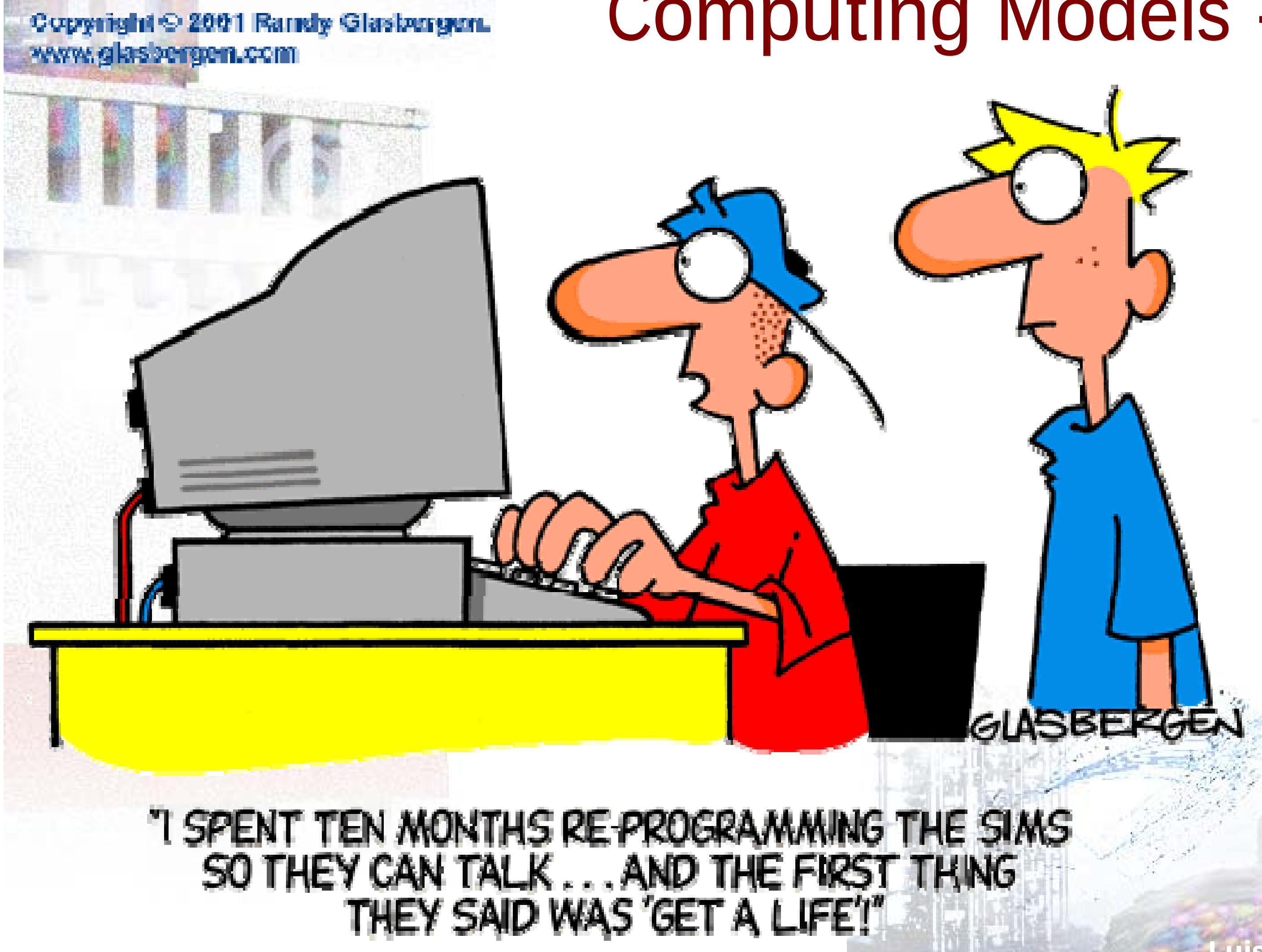
Introduction to Informatics

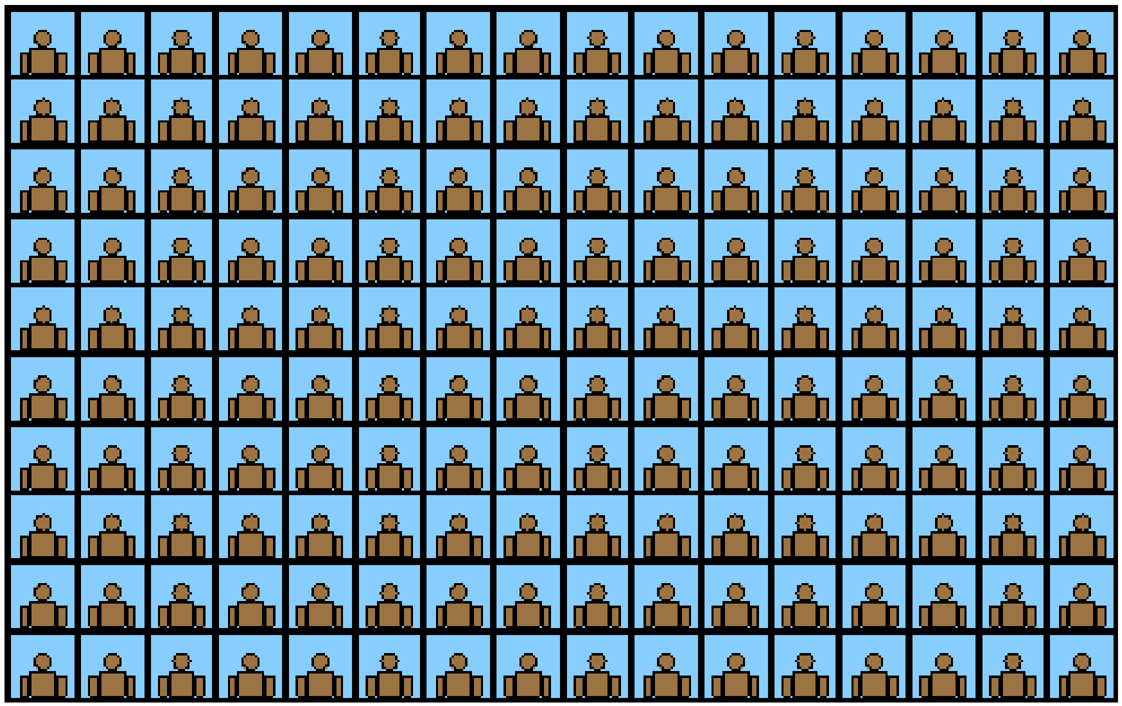
Lecture 22:

Computing Models - Algorithms



Luis M.Rocha and Santiago Schnell

NO LAB THIS WEEK !!!

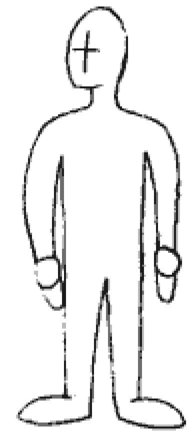


Readings until now

- Lecture notes
  - Posted online
    - http://informatics.indiana.edu/rocha/i101
      - *The Nature of Information*
      - *Technology*
      - *Modeling the World*
  - *@ infoport*
    - *http://infoport.blogspot.com*
  - From course package
    - Von Baeyer, H.C. [2004]. *Information: The New Language of Science*. Harvard University Press.
      - Chapters 1, 4 (pages 1-12)
      - Chapter 10 (pages 13-17)
    - From Andy Clark's book "*Natural-Born Cyborgs*"
      - Chapters 2 and 6 (pages 19 - 67)
    - From Irv Englander's book "*The Architecture of Computer Hardware and Systems Software*"
      - Chapter 3: Data Formats (pp. 70-86)
    - Klir, J.G., U. St. Clair, and B.Yuan [1997]. Fuzzy Set Theory: foundations and Applications. Prentice Hall
      - Chapter 2: Classical Logic (pp. 87-97)
      - Chapter 3: Classical Set Theory (pp. 98-103)
    - Norman, G.R. and D.L. Streinrt [2000]. *Biostatistics: The Bare Essentials*.
      - Chapters 1-3 (pages 105-129)
      - OPTIONAL: Chapter 4 (pages 131-136)
      - Chapter 13 (pages 147-155)
      - Chapter 5 (pages 141-144)
    - Igor Aleksander, "Understanding Information Bit by Bit"
      - Pages 157-166
    - Ellen Ullman, "Dining with Robots"
      - Pages 167-172

Assignment Situation

- Labs
  - Past
    - Lab 1: Blogs
      - Closed (Friday, January 19): Grades Posted
    - Lab 2: Basic HTML
      - Closed (Wednesday, January 31): Grades Posted
    - Lab 3: Advanced HTML: Cascading Style Sheets
      - Closed (Friday, February 2): Grades Posted
    - Lab 4: More HTML and CSS
      - Closed (Friday, February 9): Grades Posted
    - Lab 5: Introduction to Operating Systems: Unix
      - Closed (Friday, February 16): Grades Posted
    - Lab 6: More Unix and FTP
      - Closed (Friday, February 23): Grades Posted
    - Lab 7: Logic Gates
      - Closed (Friday, March 9): Grades Posted
    - Lab 8: Intro to Statistical Analysis using Excel
      - Closed (Friday, March 30): being graded
    - Lab 9: Data analysis with Excel (linear regression)
      - Due Friday, April 6
  - Next: Lab 10
    - Lab 10: Simple programming in Excel and Measuring Uncertainty
      - April 12 and 13, Due April 20



- Assignments
  - Individual
    - First installment
      - Closed: February 9: Grades Posted
    - Second Installment
      - Past: March 2: Grades Posted
    - Third installment
      - Past: Being Graded
    - Fourth Installment
      - Presented April 10th, Due April 20th
  - Group
    - First Installment
      - Past: March 9th, Being graded
    - Second Installment
      - March 29; Due Friday, April 6

Group Assignment

- Second Installment: Given the text of "Lottery of Babylon" by Jorge Luis Borges
  - Measures of central tendency and dispersion of letter frequency
  - Probability of a letter being a vowel
  - Probability of a letter being a consonant
  - Conditional probability of letters 'e' and 'u'
    - P(e|♥) where ♥ is the letter occurring before 'e'
    - P(u|♥) where ♥ is the letter occurring before 'u'
    - Compute for all letters (not space)
    - Produce histogram of P(e|♥), for all ♥.
    - Produce histogram of P(u|♥), for all ♥.
    - Discuss the independence of 'e' and 'u' from other letters
  - Upload to Oncourse

$$P(e \mid h) = \frac{|h \wedge e|}{|h|} = \frac{|'he'|}{|h|}$$

$$P(e) = \frac{|e|}{N}$$

Questions

- Over a 20-game period, the number of hits by a baseball player was
  - 1,2,0,0,1,2,2,1,0,0,4,0,1,1,3,2,1,3,0, and 1
    - Construct the Frequency distribution
    - In what proportion of games did he get at least 3 hits?
    - What is the mean, median, and mode
  - What is the line that best fits the data with the least squares criterion?
- A coin is tossed three times and an H or T (H= Head, T=Tail) is recorded each time.
  - List the elements of the sample space S and list the elements of the event consisting of
    - All heads
    - A head on the second toss
    - Two tails
  - Represent the sample space and the events above as a Venn Diagram
- One card is to be selected from an ordinary deck of 52 cards. Find the probability that
  - The selected card is an ace
  - The selected card is not a 9

Questions

- What type of Uncertainty does the Hartley measure of uncertainty measure?
- What are the units of Shannon entropy?
- Does Shannon's information theory deal with the semantics and pragmatics of a message? Please explain why?
- If we have a symbol set X={A,B,C,D,E} where the symbol occurrence frequencies are:
  - A = 0.5 B = 0.2 C = 0.1 D = 0.1 E = 0.1
    - If we know that a message is being sent in this language, what is the average minimum number of bits needed to guess the next symbol of the message?

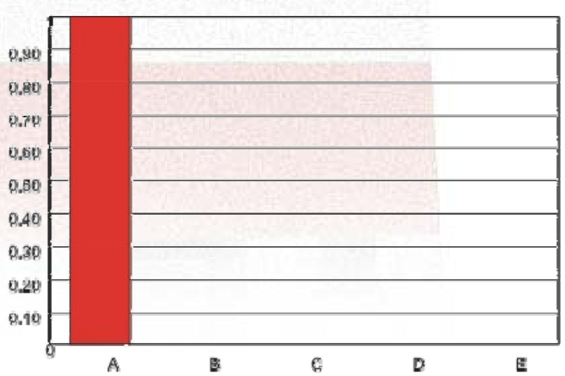
Shannon's entropy

on average, how many *yes-no* questions need to be asked to establish what the symbol is.

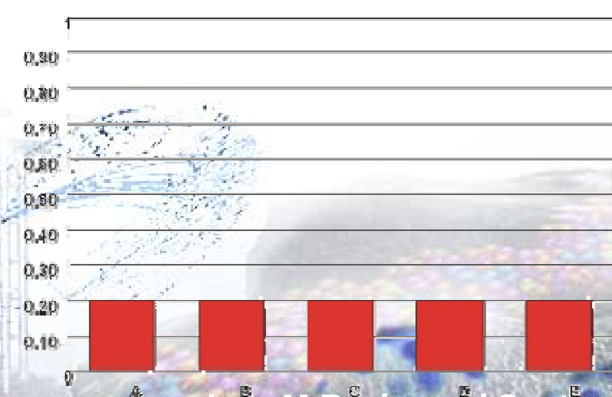
$$H_S(A) = -\sum_{i=1}^{n} p(x_i)\log_2\left(p(x_i)\right)$$

$$H_S \in \left[0, \log_2 |X|\right]$$

For one alternative



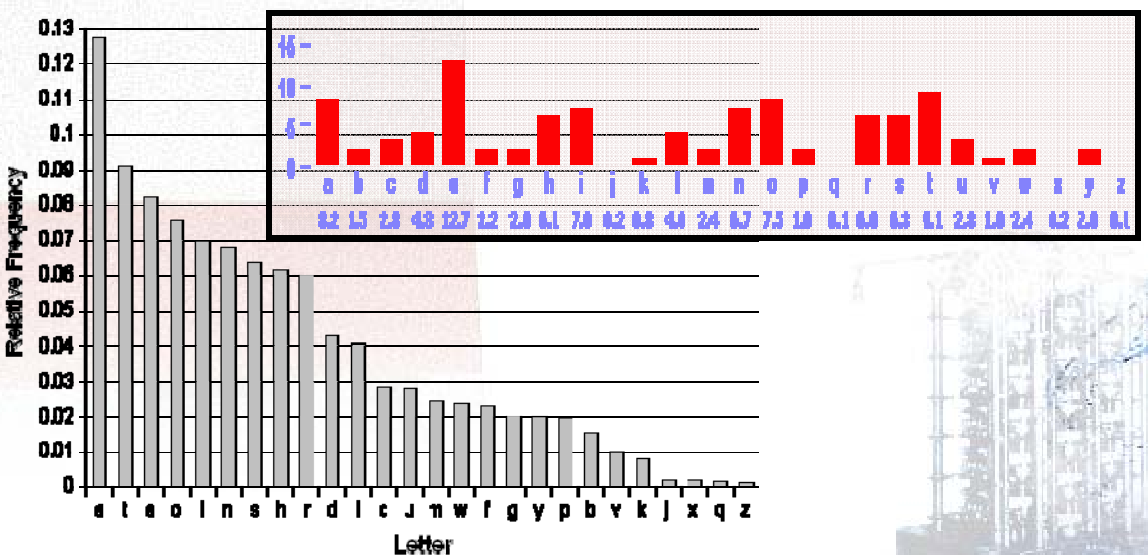
Uniform distribution



Entropy of an English Letter

- Entropy of English letter in a message
  - ***Uncertainty*** in guessing the next letter
  - ***Information*** contained in each new letter that arrives
- Assuming no word or sentence knowledge (no semantics)
  - From frequency distribution
    - H_S(letter) 4.18 bits
      - Hartley measure = log2(26) = 4.7 bits
    - How many guesses on average
- With knowledge of semantics
  - Tests with people in a sentence
    - H_S(letter) 1.1 bits
  - The value of semantics?

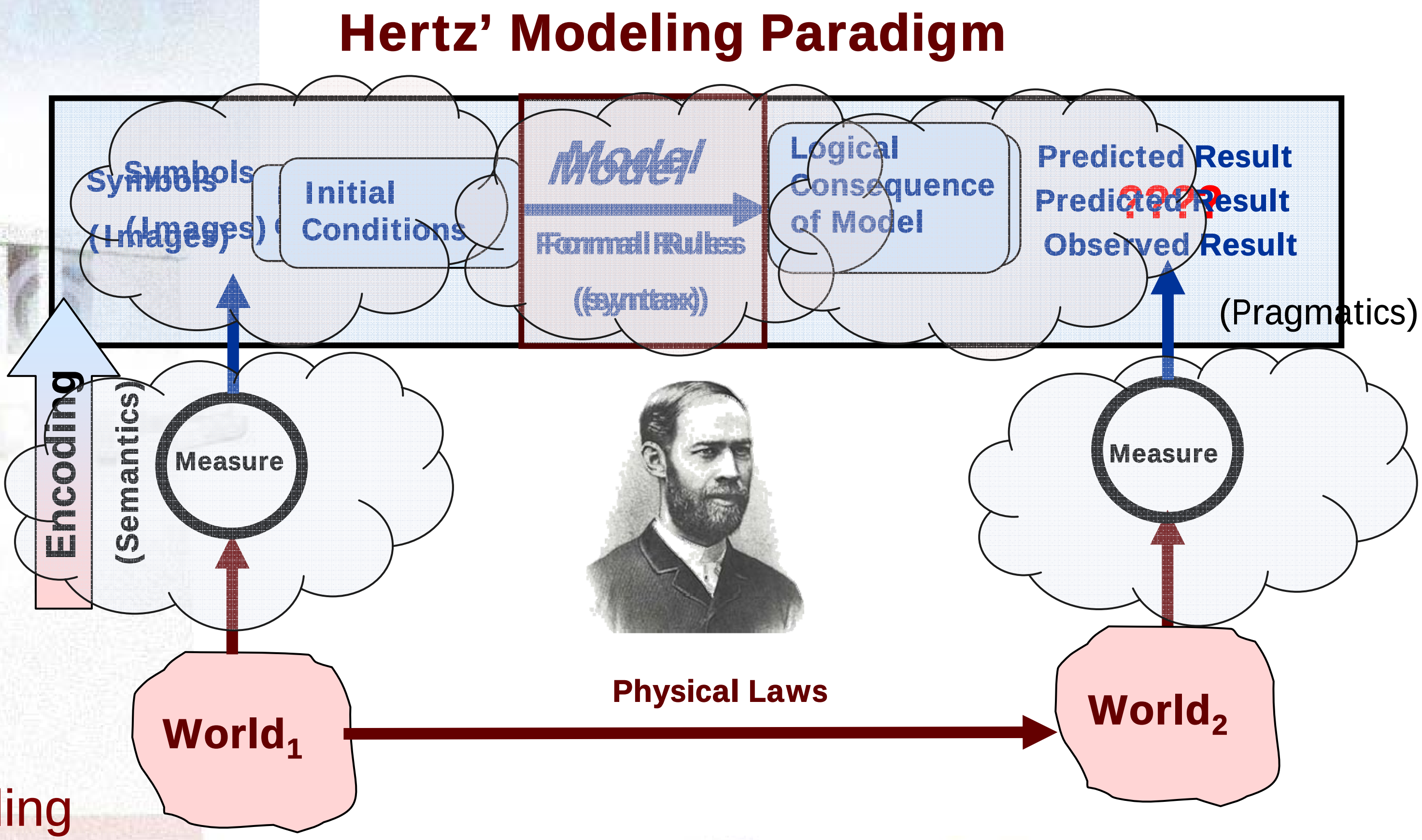
letter	estimated Probability	log_2(p)	-p.log_2(p)
a	0.0817	3.61362011	0.29522469
b	0.0149	6.06864366	0.0904213
c	0.0278	5.16877131	0.14369184
d	0.0425	4.55639336	0.19364672
e	0.127	2.9770996	0.37809165
f	0.0223	5.48681248	0.12235592
g	0.0202	5.6295009	0.11371692
h	0.0609	4.03741396	0.24587861
i	0.0697	3.84269763	0.26783602
j	0.0015	9.38082178	0.01407123
k	0.0077	7.02092584	0.05406113
l	0.0403	4.63307635	0.18671298
m	0.0241	5.37482304	0.12953324
n	0.0675	3.88896869	0.26250539
o	0.0751	3.73504328	0.28050176
p	0.0193	5.69526534	0.10991843
q	0.001	9.96578428	0.00996578
r	0.0599	4.06130019	0.24327188
s	0.0633	3.98185089	0.25203949
t	0.0906	3.46434514	0.31386967
u	0.0276	5.17918792	0.14294669
v	0.0098	6.67300264	0.06539542
w	0.0236	5.40508933	0.12755984
x	0.0015	9.38082178	0.01407123
y	0.0197	5.66588068	0.11161361
z	0.0007	10.4803576	0.00733625
			4.17623406



$$H_S(A) = -\sum_{i=1}^{n} p(x_i)\log_2\left(p(x_i)\right)$$

Luis M.Rocha and Santiago Schnell

The Modeling Relation



- Modeling
  - Compute hypothesis
- Rules from Inductive and Deductive Analysis
  - From Data analysis
  - Produce Conclusions

What’s an Algorithm?

Al Gore Rhythm ?

Algorithms

- OED
  - **Math**: A process, or set of rules, usually one expressed in algebraic notation, now used especially in computing, machine translation and linguistics.
  - **Medicine**: A step-by-step procedure for reaching a clinical decision or diagnosis, often set out in the form of a flow chart, in which the answer to each question determines the next question to be asked.
- Specifically
  - A set of instructions or procedures ***for solving a*** problem
  - For calculating or ***computing a model***.

Algorithms are like Recipes

Recipe
CHOCOLATE CAKE

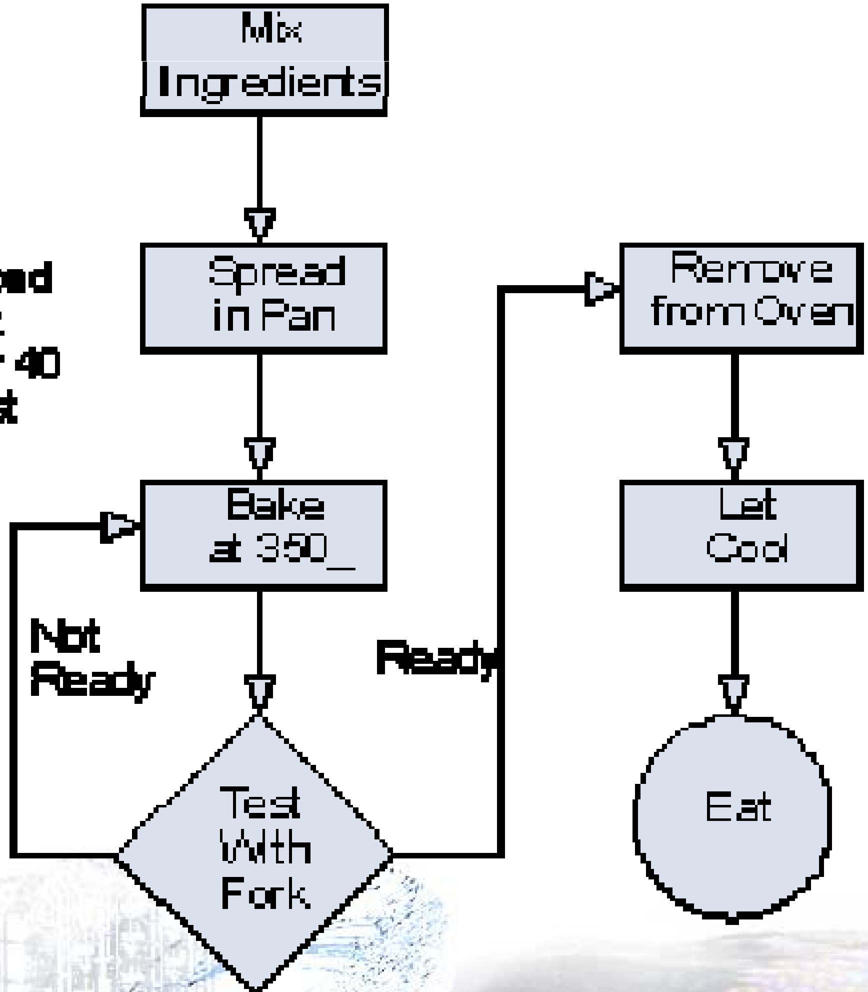
4 oz. chocolate	3 eggs
1 cup butter	1 tsp. vanilla
2 cups sugar	1 cup flour

Melt chocolate and butter. Stir sugar into melted chocolate. Stir in eggs and vanilla. Mix in flour. Spread mix in greased pan. Bake at 350_ for 40 minutes or until inserted fork comes out almost clean. Cool in pan before eating.

Program Code

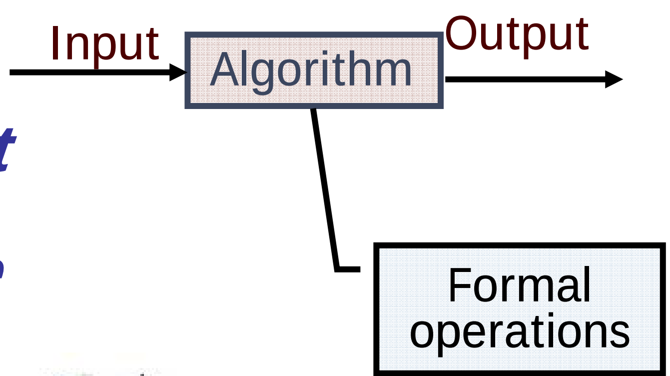
```
Declare variables:
    chocolate   eggs      mix
    butter      vanilla
    sugar       flour

mix = melted ((4*chocolate) + butter)
mix = stir (mix + (2*sugar))
mix = stir (mix + (3*eggs) + vanilla)
mix = mix + flour
spread (mix)
While not clean (fork)
bake (mix, 350)
```



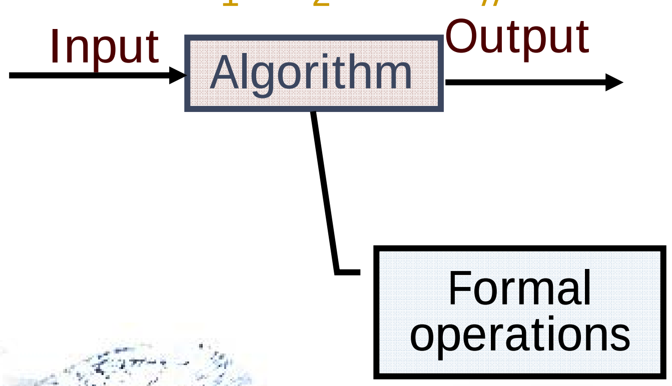
Algorithm

- Term derived from the name of the Persian mathematician Al-Khwarizmi
  - Lived in the VIII or IX century AD in Baghdad
  - Derived the concept
- In Computer Science
  - A well-defined computational procedure that takes some ***input*** values and produces ***output*** values, in a *finite amount of time* using a finite set of well-defined *operations*
  - To solve a computational problem
    - A desired input/output relationship



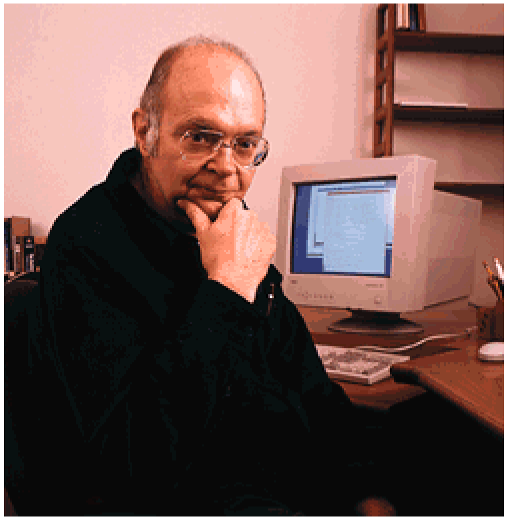
Example: Sorting

- Problem:
  - Given a random sequence of numbers, sort them in increasing order
  - Input
    - $S = <a_1, a_2, ..., a_n>$
  - Output
    - A permutation or reordering of *S*: $S' = <a'_1, a'_2, ..., a'_n>$, such that $a_1 \leq a_2 \leq ... \leq a_n$
  - Instance of the problem
    - I: <89, 54, 7, 102, 73, 15>
    - O: <7, 15, 54, 73, 89, 102>
  - Correct Algorithm
    - If for every input instance, it ***halts*** with the correct output
    - A correct algorithm ***solves*** the computational problem



Pseudocode

"*It has often been said that a person does not really understand something until he teaches it to someone else. Actually a person does not really understand something until he can teach it to a computer*." Donald Knuth



author of the *The Art of Computer Programming*, father of the field of rigorous analysis of algorithms, creator of the TEX typesetting system, etc...

- It is important that algorithms are unambiguous and precise as possible.
  - Conventions to attain layout and terminology.
- Algorithms often divided into sections
  - Input
    - the parts/components/ingredients required to accomplish the task
  - Processing
    - Actions/steps/methods to produce a result
  - Output
    - the required outcome
- Pseudo Code
  - Fake code, not really programming code
  - Specifies the steps required for processing.
  - Structured language used to specify an algorithm.

Advantage of pseudocode

- Reduced complexity
  - While writing the algorithm the developer can focus on solving the problem, not how it is written in a a particular language.
- Increased flexibility
  - Pseudo code is written so that code based on it should be able to be written in any language
- Ease of understanding
  - No need to understand a particular programming language, more like natural language
  - Employs whatever expressive method is most clear and concise
    - Even a plain English sentence

Pseudocode Statements

- Assignment
  - Used to (a) store a value in a *variable* or (b) calculating the answer to an arithmetic problem and then storing the result
  - Symbols used
    - “=“, “←”
  - Example
    - Total = 100 (storing a value)
    - Area = Length * Width (arithmetic Calculations)

Operator	Meaning	Excel	Example
()	Brackets, grouping	()	y = (a + b) * (c + d)
*	Multiplication	*	i=j*k
+	Add	+	i = i+1
-	Subtract	-	i=j-3.2
/	Real division	/	i=8/5 = 1.6
div	Integer division	Quotient (a,b)	i=8/5 = 1
Mod, %	remainder	Mod (a, b)	i=8 mod 5 = 3
ROUND	Rounds	ROUND (a, d)	i=ROUND(3.67,0) = 4
INT	Integer Part	INT	i=INT(3.67) = 3
rand	Random number	Rand() RandBetween(a,b)	i=rand(n)

Pseudocode I/O

- Input
  - Display a message asking the user for a value and store the value typed by the user in a variable.
  - Examples
    - `Input custNam`
      - displays a message asking the user to input a customers name and store the value typed by the user in the ***variable*** called *custName*.
- Display/Output
  - Displays data on the computer screen (monitor).
  - Examples
    - `Display "Width = ", width`
    - `Display "Hello World"`
    - `Display grossIncome, taxPayable`
      - Values in quotation marks are displayed exactly as stated (minus the quotation marks)
      - The values held in variables are displayed rather than the variable name.

Pseudocode Decision

- If-then-else
  - `If (letter = “a” or letter =“A”) then`
    - `display “1”`
    - `Count_A = Count_A +1`
  - `Else`
    - `Display “0”`
  - `End-if`
- Case
  - `Case letter of`
    - `“a” or “A”: display “1”, count_a = count_a+1`
    - `“b” or “B” : display “2”, count_b = count_b+1`
    - `“c” or “C”: display “3”, count_c = count_c+1`
  - `Else`
    - `Display “0”`
  - `End-case`

```
if condition then
        action
        ...
end-if
if condition then
        action1
        ...
else
        action2
        ...
end-if
case selector of
      condition1 : action1
      condition2 : action2
      condition3 : action3
end-case
```

Pseudocode Iteration or Loops

- For
  - `For x = 1 to 100 do`
    - `y=rand(100) mod x`
    - `Display y`
  - `ENDFOR`
      - Specifies exactly how many iterations to compute
- While
  - `x = 1`
  - `While ((y ≤ 4) and (x≤100)) do`
    - `y=rand(100) mod x`
    - `Display y`
    - `X=x+1`
  - `ENDWHILE`
      - The number of iterations to compute may depend on the computation itself

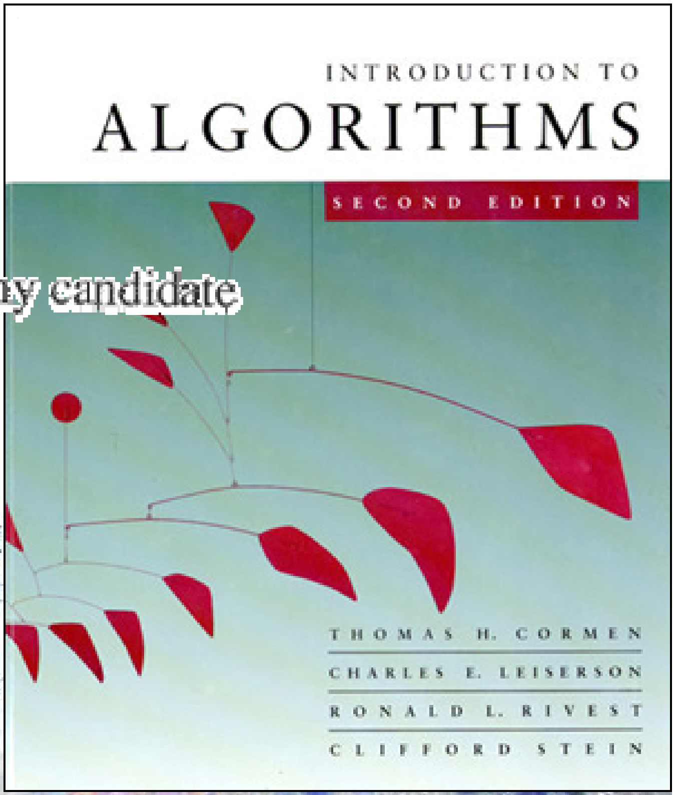
```
FOR counter = start-value to end-value DO
    statement
    statement
    ...
ENDFOR
```

```
WHILE condition DO
    statement
    statement
    ...
ENDWHILE
```

Example: Hire assistant

```
HIRE-ASSISTANT(n)
best ← 0                ▷ candidate 0 is a least-qualified dummy candidate
for i ← 1 to n
    do interview candidate i
       if candidate i is better than candidate best
          then best ← i
Hire Candidate Best
```

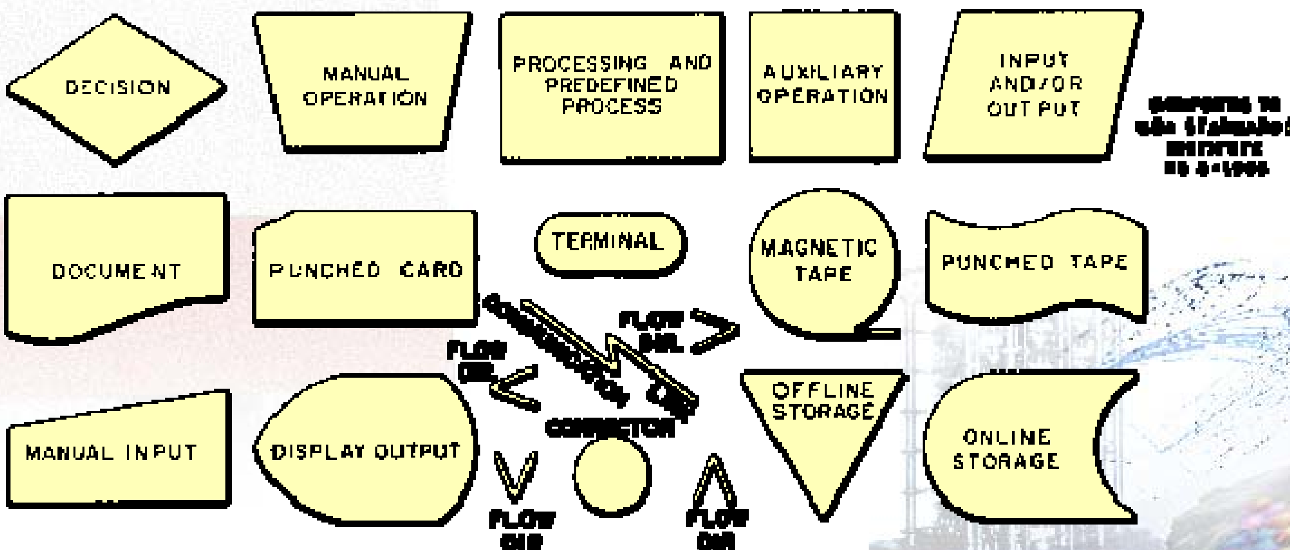
```
RANDOMIZED-HIRE-ASSISTANT(n)
randomly permute the list of candidates
best ← 0                ▷ candidate 0 is a least-qualified dummy candidate
for i ← 1 to n
    do interview candidate i
       if candidate i is better than candidate best
          then best ← i
Hire Candidate Best
```



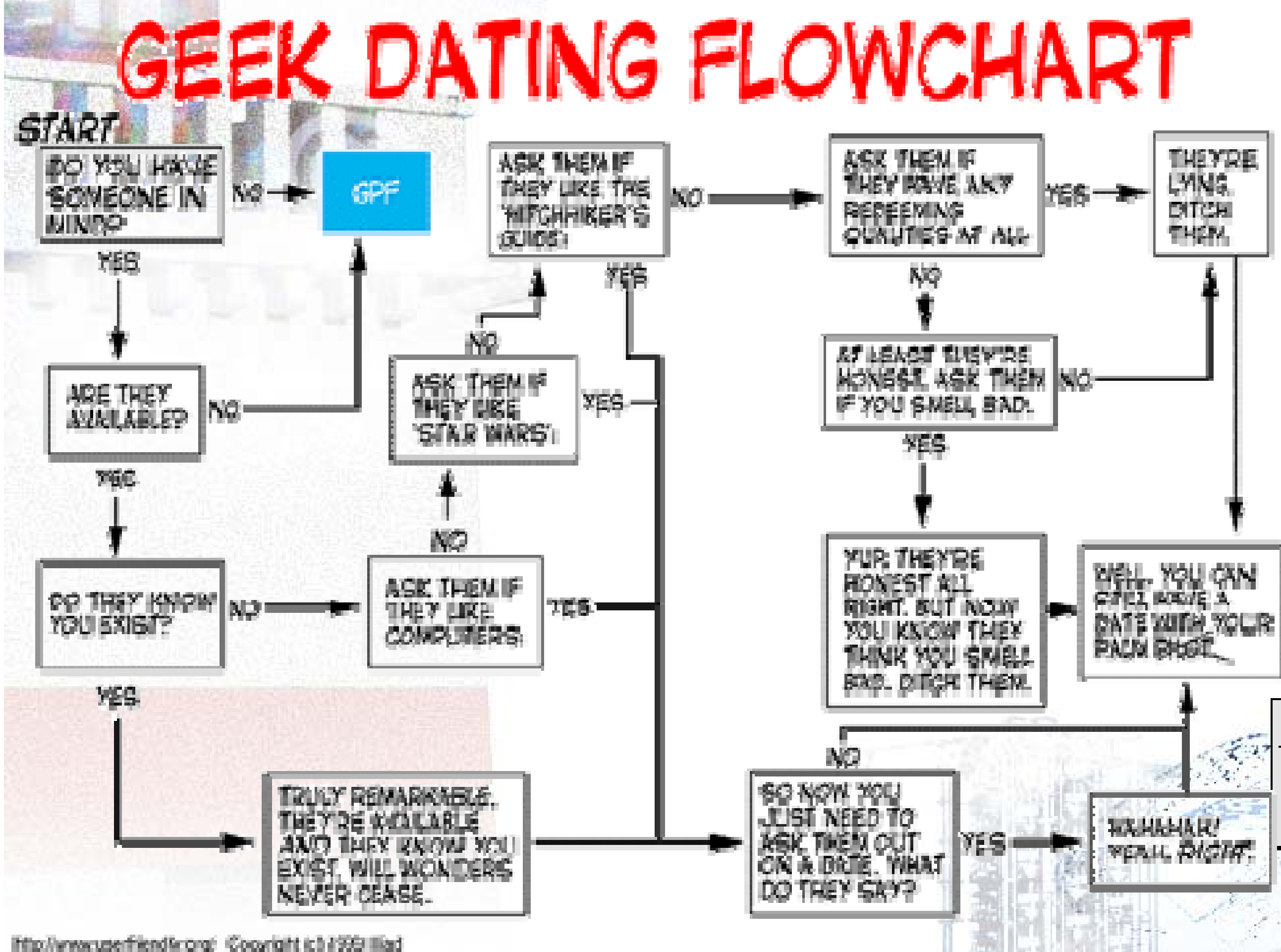
http://www.tpub.com/neets/book22/93c.htm

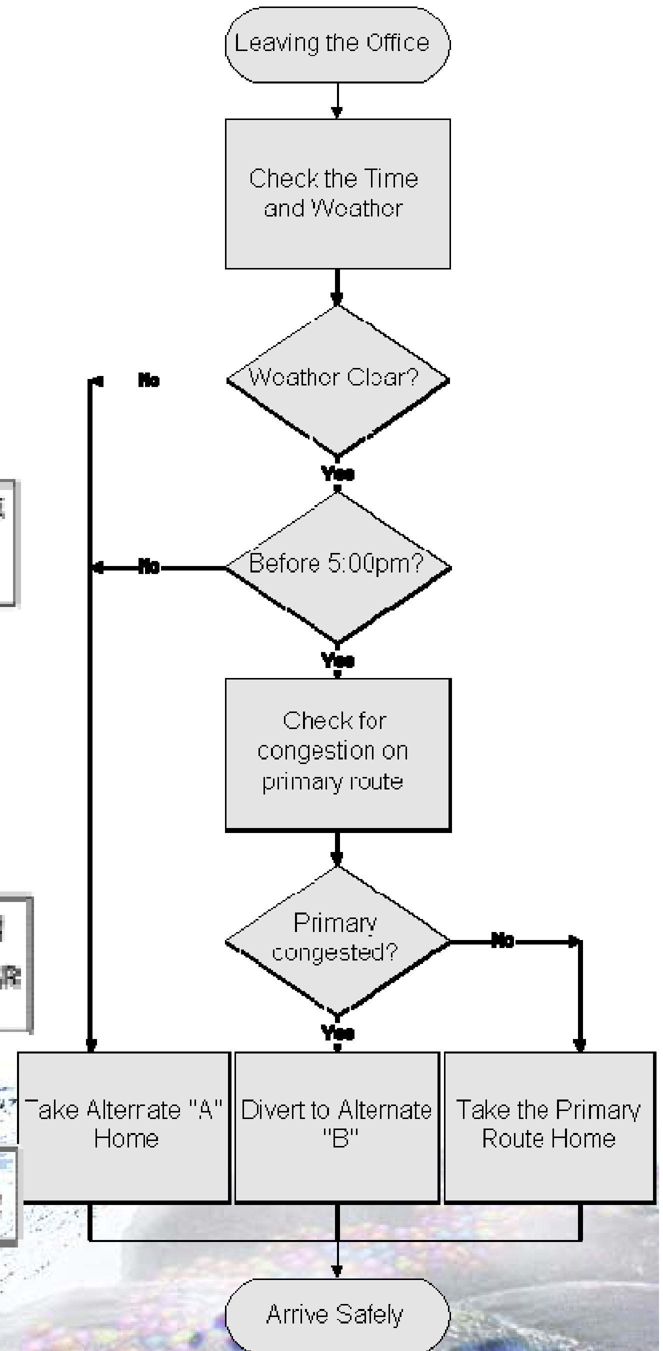
Flow Chart

- Pictorial representation of algorithm
  - Parallelogram for input/output
  - Oval for start and stop
  - Rectangle for processing
  - Diamond for decision
  - Hexagon for preparations and loops
  - Circle for connector
  - Arrow for flow direction



Flowchart Examples





http://cset.sp.utoledo.edu/sample/engt1050/engt1050_flowchart.html

Next Class!

- Topics
  - More Algorithms and Limits of Computation
- Readings for Next week
  - @ *infoport*
  - From course package
    - Igor Aleksander, "Understanding Information Bit by Bit"
      - Resources tab in onCourse.
    - Ellen Ullman, "Dining with Robots"
      - Resources tab in onCourse.
- No lab this week!!!!!!!