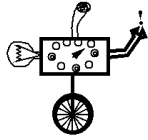


C S 6 0



Course Overview

September 5, 2001

IMPORTANT ANNOUNCEMENTS

- There will be orientation sessions for the CS computer system.
 - Held 8:30pm tonight and tomorrow, Beckman B126.
 - Attendance is *mandatory!*
- Two handouts to read, sign, and return
 - Account Request Form
 - Academic Honesty Policy

Instructor Information

Professor Chris Stone
1253 Olin
stone@cs.hmc.edu
607-8975

Office Hours:
3--5pm, Monday and Wednesday
Or, by appointment
Or, by drop-in

Text

- *Computer Science: Abstraction to Implementation* by Robert M. Keller
- Available from CS Department office (Olin 1240)
- Three options:
 - \$30 for the 2001 edition (recommended)
 - \$15 for the 1999 edition (while copies last)
 - Book chapters are available on-line (But do not print them out on HMC printers!)
- You may also want some kind of Java reference
 - But won't be needed for a couple of weeks

Computer Accounts

- You will be given an account on the main CS machine, `turing.cs.hmc.edu`
 - Important to attend the orientation meeting!
- For problems with the account, contact
 - Our system administrator
Damon Rapp, `drapp@cs.hmc.edu`
 - Or, a staff member
`staffnow@cs.hmc.edu`

Logging In

- Beckman B102 is best
- Remote login is possible, but must use ssh
 - See
http://www.cs.hmc.edu/tech_docs/qref/ssh.html
- Again, go to the orientation!

Broad Course Goals

- Exposure to a variety of important areas of computer science
- Logical thinking
- Programming practice
- Specification and problem solving

Myth

- Computer Science is the Study of Computers

"Computer Science is no more about computers than astronomy is about telescopes, biology about microscopes, or chemistry is about beakers and test tubes. Science is not about tools. It is about how we use them, and what we find out when we do." [Fellows and Parberry]

Myth

- Computer Science is "just" programming
 - Even if it were, this would still be a huge topic
 - Most CS courses do teach programming techniques along with other material.
- CS is more generally about information and computation

One Definition

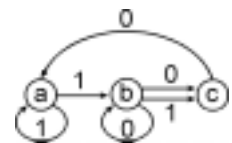
- The discipline of computing is the systematic study of algorithmic processes that describe and transform information, their theory, analysis, design, efficiency, implementation, and application.
 - Association for Computing Machinery, 1989

Course Topics (Abridged)

- Data Structures and Abstraction
- Programming Models
 - Functional Programming (rex)
 - Logic Programming (Prolog)
 - Object-Oriented Programming (Java)
 - Assembly Programming (Isca)
- Algorithms
- Grammars and Parsing
- Computer Architecture
 - Gates and Digital Logic
 - Finite state machines
- Complexity and Computability

Data Structures

- How can a computer represent data?
 - Lists
 - Sets
 - Trees
 - Graphs
 - Binary relations



Alan Perlis



"A language that doesn't affect the way you think about programming, is not worth knowing."

Sapir-Whorf



"...the real world is to a large extent built up on the language habits of the group. We see and hear and otherwise experience very largely as we do because the language habits of our community predispose certain choices of interpretation."

Programming Languages

A- - Dialect of APL used at Morgan-Stanley.
A0 or A-0 - Grace Hopper's team at Remington Rand, 1952, for the UNIVAC I
AACC - Language for building finite state automata. [?]
AADL - Automatic Architecture Description Language. *AADL: A Net-Based
ABC - (As="argument",B="basic value",C=?). Intermediate language for the ABC
ABC ALGOL - An extension of ALGOL 60 with arbitrary data structures and
ACOS - BBS language for PRODOS 8 on Apple II. Macos is a hacked version of
ACP - Algebra of Communicating Processes. *Algebra of Communicating
ACT+ - Concurrent extension of C++ based on actors. *ACT++: Building a
ACT ONE - Specification language. *An Algebraic Specification Language

... skipping 2,279 languages ...

yacc - Yet Another Compiler Compiler. Language used by the Yacc LALR
YALLL - Yet Another Low Level Language. Patterson et al, UC Berkeley.
Yay - Yet Another Yacc - An extension of Yacc with LALR(2) parsing.
Yerk - (named for Yerkes Observatory) A public domain reincarnation of
YULSP - Hewlett-Packard. A variant of Xisp for the HP-95LX palmtop.
Z+ - Object-oriented extension of Z. *Z+, an Object-Oriented Extension
ZAP - Language for expressing transformational developments. *A System for
Zed - 1978. Software Portability Group, U Waterloo. Eh, with types added.
ZENO - U Rochester 1978. Euclid with asynchronous message-passing.
ZEST - Object oriented extension of Z. *Object Orientation in Z'. S.
Zetalisp - Macisp dialect used on the LISP Machine. The many extensions
ZIL - Zork Implementation Language. Language used by Infocom's Interactive
Zipcode - Parallel language at Lawrence Livermore
zsh - Sh with list processing and database enhancements.
Zuse - named for Konrad Zuse, the designer of the first modern programming

Rex

- Little syntactic clutter
- Short, elegant programs
- Serious recursion
- Functions as first-class values
- Organized around lists
- **No assignment!**

```
// file: qsort.rex
// author: Ran Libeskind-Hadas
// purpose: illustrating quicksort

qsort(L) = qsort(L, []);

qsort([], Acc) => Acc;

qsort([F|R], Acc) =>
  Below = keep(<=<(F), R);
  Above = keep(>=>(F), R);
  qsort(Below, [F | qsort(Above, Acc)]);
```

Java

- Classes and Objects
 - Inheritance, interfaces
- Lower-level data structure implementations
 - Linked lists, stacks, ...
- Applets

Prolog

- Declarative programming via mathematical logic
- Puzzle Solving
- Simple database queries



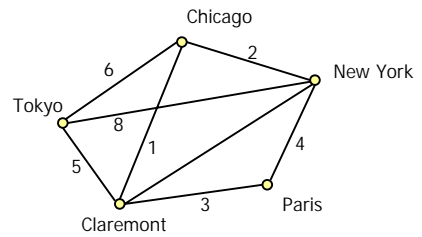
```
parent(X,Y) :- mother(X,Y).
parent(X,Y) :- father(X,Y).

grandmother(X,Y) :- mother(X,Z), parent(Z,Y).
```

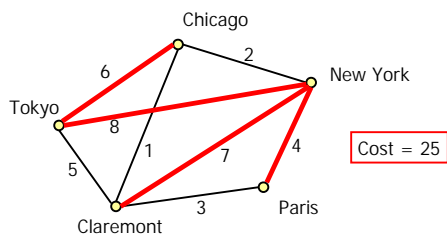
Complexity and Computability

- Generally speaking, there are three types of problems
 - Easy
 - Hard
 - Impossible

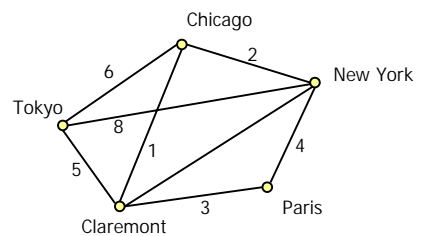
Telephone Cable Problem



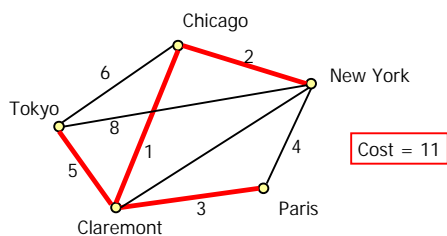
Telephone Cable Problem



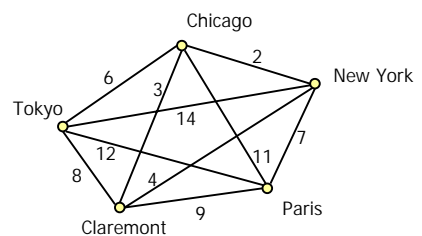
Telephone Cable Problem



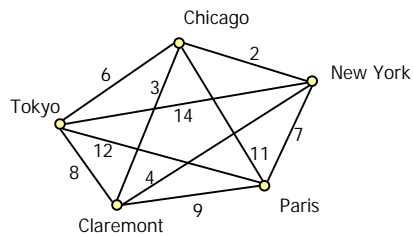
Telephone Cable Problem



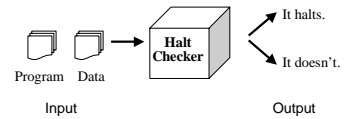
Traveling Salesperson Problem



Traveling Salesperson Problem

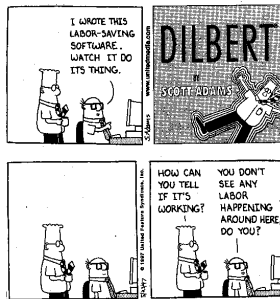


Halting Problem



Getting Help

- If a train station is where a train stops, what is a workstation?



Getting Help

- You are welcome to stop by my office to discuss problems.
- The *grutors* will also be available for help.
- You can do this in person, or send e-mail.

- It is *expected* that you will need to ask questions as a *normal* part of learning
 - In class, and while working on the assignment
 - There is no stigma attached to asking questions

E-mail Queries

- Send to `cs60help@cs.hmc.edu`
- If it is relevant, answers will be sent to the entire class.
 - Your name will be omitted unless you indicate otherwise.
- It is better to send to `cs60help` than to a specific person!

Course Expectations

- To succeed in this course, you should:
 - Read the readings
 - Work on the assignments
 - Start thinking about homework assignments immediately
 - Ask questions
 - Abide by the department honesty policy

Grading

- Approximate grading scheme
 - 50% based on homeworks
 - 25% based on final exam
 - 10% based on midterm exam
 - 15% based on class participation
- See the assignments web page for more details

Submitting Assignments

- To submit homework, log in to turing and run the program

```
cs60submit your-filename
```

- Grading breakdown is
 - 50% for general correctness
 - 25% for documentation (especially comments)
 - 25% for good approach, style, robustness

Late Assignments

- You have an automatic 1-day grade period
 - Can submit until midnight of the day after the due date.
- Beyond this, late work accepted only with a note from your Dean of Students

Tips for Assignments

- Start early!
 - Then you can walk away if you get stuck
 - Let your subconscious mind do some of the work
- If you do get stuck, ask questions before you spend hours banging your head on the wall

Web Pages

- The CS 60 home page is:


```
http://cs.hmc.edu/courses/current/cs60
```
- Includes lots of useful and/or important information
- Will be constantly updated as the course proceeds.