CSE 111

Fall 2010

November 29 – December 3

Announcements - Lab

- Lab 7 this week and next week in lab sessions. You must attend both lab sessions or you will receive a grade of zero for Lab 7.

Announcements - Other

- Lecture Schedule:
- Today, Friday, (some of) Monday: Finish up lecturing on last section of course.
- Some exam review on Monday 12/6
- Wednesday 12/8 – Exam 4
- Friday 12/10 – Return exams and any other papers – last day to collect papers – last day of lecture.

Functions

All programs can be viewed as functions.

- They take an input
- Do some computation
- Produce output

If we can derive an algorithm / build a program, we say the "~~problem~~ function" is computable.

If we want to determine computability, we use a hypothetical machine (Turing machine)

Consists of:

- control unit that can read & write symbols to a tape
- tape is infinitely long in both directions divided into cells

Each cell contains only one symbol

- The set of symbols for a Turing machine is called an alphabet

When the Turing machine computes:

- Read the symbol in the current cell
- Process the symbol
- Write a (new) symbol in the cell
- Move the head ~~the~~ to the left or right

Noncomputable Functions

The Halting Problem

Is there a way to determine if a program will halt (terminate/end) if started under certain conditions.

Proof that the Halting Problem is not computable

Assume that there is a program that solves the halting problem.

- The user inputs a computer program on screen 1 & clicks "Next"

- The user enters an input on screen 4 clicks "Next"
- Screen 3 displays "Halt" if the program halts on the input and displays "Doesn't Halt" if it doesn't

can't happen - so we can't write this program

We know, we can solve it - can we solve it efficiently?

Analyze program structure to determine how long the program takes to compute.

→ loops

P vs. NP

↓

Class of problems that run in "polynomial time"

$f(n) = n^2 + 2n + 5$

size of input to the program

NP

↓ class of problems that are not solvable in polynomial time.

$f(n) = 2^n$

If we know a possible solution, we can verify that it is or is not a solution in polynomial time.

Traveling Salesman Problem

Salesman must visit all his/her clients in various cities without exceeding the travel budget. Find a path starting at home, visiting each city, and returning home without exceeding budget (mileage).

All problems in P are in NP.

Open Question:

Are all NP problems in P?

Ethical Issues in Computing

- What are the responsibilities of computing professionals?
- What are the responsibilities of computer users?

Privacy

- What information do you think is important to be private?
- What are the expectations of privacy in today's society? How have computers changed that?
- What do sites like Facebook do to block certain elements of privacy?

VIDEO FROM MONDAY 12/6

- http://www.youtube.com/watch?v=rbzJTTDO9f4&feature=player_embedded