Algorithms. Key components of algorithms: sequence, conditional, iteration, recursion
Class website

- http://www.soe.ucsc.edu/classes/cmps010/Spring11/
- Please write this down, and bookmark it

- Holds:
  - Syllabus (including homework due dates)
  - Homework assignment descriptions
  - Description of course readings
  - Links to class lecture notes

- The final exam is scheduled for Tuesday, June 7, 8am-11am
  - This class will have a final exam. Please plan on this.
Tutoring available

- Learning Support Services (LSS)
  - Has tutoring available for students in CMPS 10
  - Students meet in small groups, led by a tutor
  - Students are eligible for up to one-hour of tutoring per week per course, and may sign-up for tutoring at [https://eop.sa.ucsc.edu/OTSS/tutorsignup/](https://eop.sa.ucsc.edu/OTSS/tutorsignup/) beginning April 5th at 10:00am.
  - Brett Care - [bcare@ucsc.edu](mailto:bcare@ucsc.edu) is the tutor for CMPS 10 that LSS has hired
DRC Students

- If any student in the class requires a special accommodation for test taking or other assignment, please contact me
  - In person, or via email, ejw@cs.ucsc.edu
  - If you don’t contact me, I will not know you need this accommodation
  - The DRC office no longer sends notifications out about this
Homework #2

- Now due next week in class on Wednesday
- A series of questions asking you to create class models for different physical world situations (shown in photographs)
  - These are similar to the examples we have done in class
- Other questions asking you to perform operations on basic data structures
  - E.g., push and pop on a stack; enqueue and dequeue on a queue
  - These are similar to examples shown in class notes

- No lecture on Friday
  - But, TAs and tutors will be available in class to help you with this homework assignment
- Help section
  - Tuesday, 3-5pm, Engineering 2, room 307
  - Drop-in help on this homework assignment
Please see me

- Would the following students please see me at the end of class
  - I need email addresses to add to eCommons
    - Katherine Kupis
    - Chris Esposito
    - Ileena Mitra

- Thank you!
What is an algorithm?

- Computers are capable of performing many complicated functions
- However, at the end of the day, a computer must be told exactly what to do
  - It does not have free will, in the same way as humans, animals, insects, etc.
  - Computers must be provided a very detailed sequence of instructions

- This notion of a sequence of instructions is called an algorithm

- An algorithm is a list of instructions for performing a specific task, or, for solving a particular type of problem
Intuitive example of an algorithm

- Recipes are examples of algorithms
- Chocolate Chip Cookies

**Directions**
1. Preheat oven to 350 degrees F (175 degrees C).
2. Sift together the flour and baking soda, set aside.
3. In a large bowl, cream together the butter, brown sugar, and white sugar.
4. Beat in the instant pudding mix until blended.
5. Stir in the eggs and vanilla.
6. Blend in the flour mixture.
7. Finally, stir in the chocolate chips and nuts.
8. Drop cookies by rounded spoonfuls onto ungreased cookie sheets.
9. Bake for 10 to 12 minutes in the preheated oven, until edges are golden brown.

**Ingredients**
- 4 1/2 cups all-purpose flour
- 2 teaspoons baking soda
- 2 cups butter, softened
- 1 1/2 cups packed brown sugar
- 1/2 cup white sugar
- 2 (3.4 ounce) packages instant vanilla pudding mix
- 4 eggs
- 2 teaspoons vanilla extract
- 4 cups semisweet chocolate chips
- 2 cups chopped walnuts (optional)

allrecipes.com//Recipe/award-winning-soft-chocolate-chip-cookies/Detail.aspx
Aspects of the Chocolate Chip Cookie Algorithm

- Has a **sequence of steps**
  - Each step should be performed in order
  - Do step 1, then 2, then 3, etc.

- Has a **loop** for baking
  - Do:
    - Keep cookies in oven
  - Until:
    - Cookies have golden brown edges and at least 10 minutes have passed since cookies were placed in the oven
    - This is a **condition** for ending the loop

- **High-level operations**
  - Requires a medium level of cooking knowledge to understand steps
    - Does “Blend” in step 6 mean use a blender?
    - “Stirring” in the eggs in step 5 has an implicit precondition of cracking open the eggs and removing the shell
    - What does it mean to “drop” cookies in step 8?

- Natural language algorithms traditionally are full of this kind of implicit knowledge and ambiguity

**Directions**

1. Preheat oven to 350 degrees F (175 degrees C).
2. Sift together the flour and baking soda, set aside.
3. In a large bowl, cream together the butter, brown sugar, and white sugar.
4. Beat in the instant pudding mix until blended.
5. Stir in the eggs and vanilla.
6. Blend in the flour mixture.
7. Finally, stir in the chocolate chips and nuts.
8. Drop cookies by rounded spoonfuls onto ungreased cookie sheets.
9. Bake for 10 to 12 minutes in the preheated oven, until edges are golden brown.

allrecipes.com//Recipe/award-winning-soft-chocolate-chip-cookies/Detail.aspx
Key building blocks of algorithms

- Algorithms are comprised of the following building blocks
  - Operations
    - Individual steps to perform
  - Sequences of operations
    - These are sequences of steps, one after another
  - Condition
    - A particular state of the world while the algorithm is operating
    - Example: “edge of cookies is golden brown”
  - If .. Then ... Else
    - If *some condition* do ... else do ...
  - Iteration
    - Repeating a sequence of steps until some condition holds
  - Recursion
    - Repeating a sequence of steps by re-starting the same sequence of steps with new input conditions, repeatedly

- Watch:
  - Algorithmic Thinking: Knight School 2009
    Examples of students acting out sorting algorithms
    http://www.youtube.com/watch?v=INHF_5RlxTE
Operations

- Actions performed inside the algorithm
  - The basic building blocks of any algorithm
- In the recipe example these were actions such as
  - Stirring, mixing, beating, creaming, sifting, baking, etc.
    - These are relatively high level operations
- For computers being instructed via a programming language, each operation is simple:
  - Assignment
    - Giving a variable a value
    - Temperature = 49.5
  - Performing an simple computation
    - Addition, multiplication, division, log, square root, etc.
    - Area = 2 * 3.14159 * radius
  - Telling another function to perform
  - Moving data
  - Allocating memory
  - Read item of input
  - Produce item of output
Sequences of operations

- A sequence of operations is:
  - A set of basic operations…
  - Performed one after another.

- Each operation is performed and completes before the next one begins.

- Operations need to be performed in the order they are written, (from top to bottom when written out for humans to read)
  - Each operation potentially modifies the state of the situation and may depend on the current state at the start of the operation
  - Example: for chocolate chip cookies, adding in the chips changes the batter, by adding the chips
    - If this is done too early, some of the dry ingredients will stick to the chips
    - The sequence of the operations matters

- This is the von Neumann fetch-execute cycle all over again
Condition

- Most algorithms require their behavior to change based on the current state of the situation
  - For computers: a change based on the current values of variables inside the computer

- Takes the form of a comparison
  - Oven Temperature is equal to 350 degrees
  - The color of cookie edges is equal to golden brown

- Conditions are used to determine how an algorithm should branch
  - That is, how the sequence of steps to be performed should be changed
If *condition* then ... else ...

- Based on a condition, perform one set of operations
- If the condition does not hold, perform a different set of actions

- If *car gas tank is not full* then
  - keep pumping gas

else
  - stop pumping gas
  - collect receipt

\[ \text{Gas tank is full} \]
Iteration

- Perform a sequence of actions repeatedly
  - Until a condition becomes true
    - **For** *(until condition)* ...
  - While a condition is true
    - **While** *condition* …
    - **Do** … **while** *condition*

- Each time through the sequence of steps is an **iteration**

- Example
  - **While** *the color of cookie edges is not equal to golden brown*
    - Cook cookies another 30 seconds
  
  OR

  - **Do**
    - Cook cookies another 30 seconds
    - **While** *the color of cookie edges is not equal to golden brown*
Recursion

- When creating algorithms that operate on trees
  - It can be more convenient to describe what the algorithm does at each level
  - Then repeat the algorithm at each level
- Example:
  - Are you related to George Washington?

- General approach
  - \texttt{Is\_Related\_to\_Washington(Person P)}
    - Determine the parents of P. These are Parent1, parent 2, ... parent N
  - Is any parent George Washington?
    - Yes! Person is related, print this out and stop
    - No. Repeat check for all parents of parents
      - That is:
        - \texttt{Is\_Related\_to\_Washington(Parent 1)}
        - \texttt{Is\_Related\_to\_Washington(Parent 2)}
        - ...
        - \texttt{Is\_Related\_to\_Washington(Parent N)}

- The description of \texttt{Is\_Related\_to\_Washington} depends on itself!
  - That is, it is \textit{recursive}
Key building blocks of algorithms

- Using these key building blocks, it is possible to describe most algorithms executed by a computer, and encountered in the real world as well
  - Operations
    - Individual steps to perform
  - Sequences of operations
    - These are sequences of steps, one after another
  - Condition
    - A particular state of the world while the algorithm is operating
  - If .. Then ... Else
    - If *some condition* do … else do …
  - Iteration
    - Repeating a sequence of steps until some condition holds
  - Recursion
    - Repeating a sequence of steps by re-starting the same sequence of steps with new input conditions, repeatedly

- Relating to the last lecture, every computer algorithm describable using these building blocks can also be described using a Turing machine.
  - That is, these are Turing-equivalent
What kind of knowledge is an algorithm?

- Some of the most important contributions made by Computer Science are algorithms
  - That is, efficient ways of performing certain kinds of tasks
  - We saw an example of sorting earlier

- One can think of finding an algorithm as a kind of discovery
  - So one could think of these as scientific knowledge

- On the other hand, they’re very theoretical constructs
  - Perhaps they’re more like mathematics?

- One thing for sure, it is very valuable to discover an important new algorithm
  - Google, for example, was founded based on the advantages granted by their Page Rank algorithm for listing web pages in search engine results
In-class quiz

- OK, as threatened multiple times, today we’re having our first in-class quiz

- The general idea is that if you’ve been regularly attending lecture, you’ll do very well

- Please take out a blank sheet of paper
  - I can provide you one if needed

- Please wait until at least 5 minutes have passed before getting up and leaving
  - This provides a good quiz-taking environment for others

- Times when you can leave will be announced

- Closed book, closed notes
1. Give the first and last name of the person who invented the Turing Machine.
   - Alan Turing

2. How many physical, real-world Turing Machines did their inventor ever build?
   - None

3. What part of a Turing Machine has infinite length?
   - The tape

4. True or false: All numbers are computable.
   - False. There are more uncomputable numbers than computable ones

5. True or false: A Turing Machine can represent any feasible mechanical computation
   - True. That is one of the key contributions of the Turing Machine

6. Give two basic data types that are used to represent numbers
   - Integer (int), float (could also be character, char, double, etc.)

7. True or false: Post’s Correspondence Problem can be solved for an arbitrary set of cards
   - False. This was the example provided in class of an uncomputable problem.

8. The chocolate chip cookie recipe is an example of what?
   - An algorithm. Also: a sequence of steps… (i.e., the definition of an algorithm), or a natural language algorithm