Curriculum Map

# 2.1 Data Modeling

## Activities

## Essential Questions

* What are techniques for representing real world data

## Skills

* Mapping real-world concepts into data structures

# 2.2 Problem Solving

## Activities

* Life problem

## Essential Questions

* What are techniques for problem solving

## Skills

* Applying problem solving skills to solve larger problems

# 2.3 Running Time

## Activities

* Measuring performance
  + Measuring the running time of O(1), O(N), and O(n^2) algorithms
  + Graphing the results in a spreadsheet
* Using parallel processing to speed something up
  + Web workers

## Essential Questions

* How is software performance measured abstractly?
* What is Big “O” notation? Why does it matter?
* What is a CPU? What is a core?
* How is the speed of a CPU measured?
* What are units used in measure CPU performance?
* What is parallel processing?
* When does performance matter?

## Skills

* Recognizing Big-“O” running time of simple algorithms
* Using graphing to understand the difference between running times
* Measuring software performance
* Using a profiler to find performance problems
* Making performance improvements
* Using parallel processing to improve performance

# 2.4 Memory Usage

## Activities

* Understanding memory size
  + Pieces of paper on the wall
  + How far do they stretch? Hawaii? Around the world?
* Measuring memory usage
  + Using Activity Monitor / Task Manager

## Essential Questions

* What is memory?
* What are memory units?
* How is information represented digitally?
* What is a bit? What is a byte?
* How are numbers written in base 2 (binary)?
* How is a number represented?
* Are there limits on precision?
* How is boolean value represented?
* How is text represented?
* How is an array represented?
* What are bitwise OR, bitwise AND, and bitwise NOT?

## Skills

* Measuring memory usage
* Estimating memory usage
* Making memory usage improvements
* Using bitwise operators

# 2.5 Recursive functions

## Activities

* Mountain range
* Concentric circles

## Essential Questions

* What is recursion and what is it used for?
* What is the role of the base case of a recursive algorithm?
* What are some pitfalls of recursion?
* Why is recursion useful?

## Skills

* Breaking down a problem into sub problems.
* Drawing lines and circles using canvas.

MAY NOT HAVE TIME FOR THE FOLLOWING

# Testing

## Essential Questions

* What are different types of testing and what are their goals?

## Skills

* Using regression testing
* Using integration testing
* Using unit testing

# Binary Search

# Sorting

## Essential Questions

* What is sorting?
* What is a merge sort?
* What is an insertion sort?

## Skills

* Using array sort
* Using a comparer

ADDITIONAL NOTES

MODELING DATA

two types of values:

- scalars: numbers, text, booleans, null

- structures: objects (related values), arrays (repeating values)

- we can combine these arbitrarily

modeling data

- given a problem, what data structure should I use?

- what do I want to represent (not everything)?

- how should I represent it?

practice (problems)

-

PROBLEM SOLVING

- simplified version of "how to program it"

life algorithm

- http://en.wikipedia.org/wiki/Conway%27s\_Game\_of\_Life

- 2d or 1d array?

- different seeds

- requires array

- requires animation

- have all the grids as data

bar chart

- plot the number of students in a school object

- make an array with the values in it

- generalize this to take an array of values

- need to scale the values (min, max)

RUNNING TIME

Number of items. O(1)

Sum of items. O(n)

Selection sort. O(n^2)

Binary search O(log n)

Merge sort. O(n log n)

Generate an array with n random numbers.

Test with 1,100,1000,10000,100000,1000000.

Test function: will generate array for N, and call other function with array, and return time. We can get time from debugger, and chart N vs. Time.

What do we see? Different curves: quadratic, logarithmic, linear, constant.

Introduction to asymptotic notation, and classes of algorithm running times.

Notice that we all got the same result, even though our computers were different speeds.

Why running time is important. Explanation of what is computable by different running times.

SORTING

sorting an array

- bubble sort

- light ones float to the top

- repeat with smaller length list

- http://cs50.tv/2012/fall/shorts/bubble\_sort/bubble\_sort-720p.mp4

- mentions asymptotic notation

comparer

- generalizes the notion of comparison

insertion sort

- http://cs50.tv/2012/fall/shorts/insertion\_sort/insertion\_sort-720p.mp4

- does insertion at the end by shifting to the right until correct position found

selection sort

- http://cs50.tv/2012/fall/shorts/selection\_sort/selection\_sort-720p.mp4

- pick the one that is smallest, of the remaining

merge sort

- nlogn

- requires additional storage

javascript methods

- quicksort

- sort method on array

SEARCHING

linear search

- http://cs50.tv/2012/fall/shorts/linear\_search/linear\_search-720p.mp4

binary search

- first part: binary search in array

- http://cs50.tv/2012/fall/shorts/binary\_search/binary\_search-720p.mp4

ADDITIONAL DATA STRUCTURES

stack

- use insert/delete to implement a stack

- concept of fifo, real world equivalents

queue

- use insert/delete to implement a queue

- concept of lifo, real world equivalents

hash tables

- the concept of hashing

- TODO: has to be a simple example for this

trees

- binary search tree?

graphs

- facebook friends