Announcements

- Welcome to the last class!!
Today

- Problem solving
- Data structures take home lesson...
- Review...
- Sample questions...
Undecided problems

- Can a computer solve every problem?
Example:

Given a set of numbers:
- \{-2, -3, 15, 14, 7, -10\}

Can we find a subset which will add up to some number, example 0?
Theoretical computer science

- Certain problems can be solved in polynomial time
- Certain problems can be verified in polynomial time
- Those problem which are run on non deterministic Turing machine, can they be solved in P time ??
  - Million dollar prize
  - P $\equiv$ NP
Undecided problems...

Example:
- Write a program to predict if a program will infinite loop
Loop checker...

- How does it check itself?

- Example if it has to check a program P, if P infinite loops, prints out message, else goes into infinite loop 😊

- Now can we run such a program?
Classes of problems

- Polynomial problems
- Nondeterministic polytime problems
- Traveling salesman problem.
Take home lessons

- Need to think about the problem
- Decide on DS approach
- Test solution
  - Might not translate well
  - Might introduce errors (see if runtime doesn’t match)
  - Try another DS
REVIEW!!!
basics

- should remember your fundamental data structures
  - lists
  - queues
  - stacks

- form the basis for advanced structures
  - trees
  - hash tables
  - graphs
You are familiar with runtime concepts

Implementation can make break a data structure
- Arrays, linked lists affect on implementation

Coding ideas we’ve covered
- Recursion
- Code layout
- Efficiency concerns
hash tables

- idea of hash table
- hash functions
- collision resolution
- scalability issues
- Example: bloom filter hashing

- in use with other DS
Trees

- implementation details
- BST
- Traversals
- AVL
- B+ trees

- how are trees related to graphs ??
- **sorting**
  - Basic sorts
    - bubble
    - selection
    - insertion
  - Better
    - heapsort
  - Even Better
    - mergesort
    - quicksort
- analysis of sorts
sets

- idea of sets
- union / find
- path compression
- union by rank
Graphs

- implementation
- sorting
- shortest paths
- network flows
- MST
- Expansion
  - DFS
  - BFS
- Algorithms
Algorithms

- analyzing general run times
- comparing classes of algorithms
- getting a feel if it’s a good algorithm
Lessons Learned

- Think then code!

- Try not to reinvent wheel or do things in $n^3$ or $n!$ land 😊
practice I

- say you are standing in front of a wall which stretches infinitely to the left and right.
- there is a door somewhere N steps away, but you don’t know which directions
- give an algorithm to find it

- now give a linear algorithm and exact runtime!
for prim’s algorithm...if you have a choice between two edges, can you explain how it makes no difference which one is chosen
Practice III

- You are given an array that contains $N$ numbers.
- We want to determine if there are two numbers whose sum equals a given number $K$.
- For instance, if the input is $8, 4, 1, 6$ and $K = 10$, the answer is yes (use 4 and 6).
- A number may be used twice.
- Do the following:
  - Give an $O(N^2)$ algorithm to solve this problem
  - Give an $O(N \log N)$ algorithm to solve this problem.
Practice IV

- Is it possible to implement insertion sort for sorting linked lists?
- Will it have the same $O(n^2)$ running as an array version?
conclusion

- hope you have fun learning about the theory and programming of DS and Algorithms
- please remember its open your notes and book but nothing else (no random websites on your sneaker browsers..)
- please remember to fill out the online survey for the class on courseworks...
  - Worth 5 points on final
where to next

- so what can you do now 😊
Thank you

good luck on the final....
Random question time…

- If you were to create a tic tac toe player and the computer would use a tree structure to represent all possible moves, what would the branching factor of the tree be and what would be the height of the tree assuming you represented every possible move?
Random puzzle

- You fall asleep in your data structures course and find yourself in another class.
- They are trying to stress test various kinds of glass jars to determine the height for which they can be dropped and still not break.
- They have a setup where they have a ladder leaning against the wall with n steps. You want to find the highest step which the glass jar will survive a fall (call it the HSR = highest safe rung).
- Clearly they have no idea where to start...lucky you know data structures and algorithms.
  a) describe an algorithm which takes n jars to find the correct HSR
  b) now describe an algorithm which takes log n jars
  c) what is the best algorithm you can think of?