Week 1

9/27/21

- Where did C come from?
  come from B by Ken Thompson

- int main(void)
  - exactly one main

- { }
  - used to group statements.
  {} called a blocks
- if do not specify type, default to int type

- While : Top test loop
- for loop also top test loop

"#" not apart of C.

Python interpreted language
- meaning excutes code without compilers.

Compilier VS interpreter

faster

Lec 2

7/29

∧ (and)	T	F
T	T	F
F	F	F

∨ (or)	T	F
T	T	T
F	T	F

$A \vee (B \vee C) = (A \vee B) \vee C$

$A \vee B = B \vee A$

$A \vee (B \wedge C) = (A \vee B) \wedge (A \vee C)$

¬ (no)	
T	F
F	T

$A \wedge \neg A = 0$

$A \vee \neg A = 1$

⊕ (XOR)	T	F
T	F	T
F	T	F

Have 1st or 2nd not Both

$A \oplus A = 0$

$A \oplus 0 = A$

$A \oplus \neg A = \neg A$

In C 0 is false

In C

&& (and)

|| (or)

loops

repeat sequence of code.
Don't cross other loops.
- Go to does cross.

Bottom test loops

- do{} while()

Top tested

for,
while

CSE13s

Lec 3

Functions return things.
if not, are void function

Function in programming is a block of code that performs a specific task.

Functions defined Once
declared before use
Called many times.

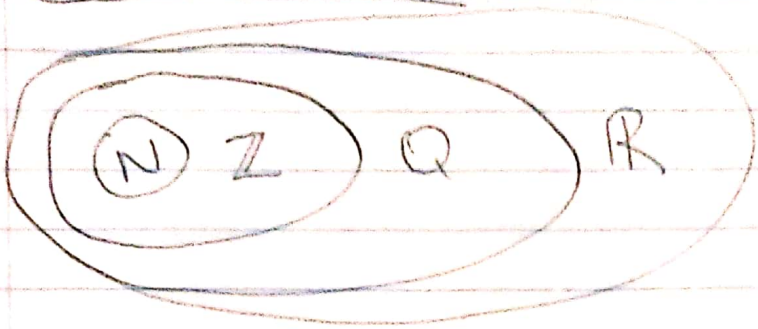
Can't return a array

C uses call by value, except for arrays.

Addresses, instead of values, are passed as arguments.

Lec 4

Kinds of Numbers



natural number := $\{0\} + \mathbb{Z}^+$

Every number in base 10, written as a polynomial

$1962 = 1 \times 10^3 + 9 \times 10^2 + 6 \times 10^1 + 2 \times 10^0$

License Plate are numbers written in base 36

unsigned { short, long, long long } int

unsigned { char } ← 8 bit int

In C

short	Usually 16 bits
int	At least 16 bits
long	At least 32
long long	Probably 64 bits

Suppose we have k bits, then we can have every positive integer from 0 to $2^k - 1$

2's complement

flip +1

0001 → 1110 → 1111

Floating Point numbers

$\mathbb{F} \subseteq \mathbb{R}$ not real

$\mathbb{F} \subseteq \mathbb{Q}$ not rational

$\mathbb{F} \subseteq \mathbb{Z}$ Approximations

Single Precision

Double Precision

Lec 5

- Computers can only do basic operations such as add, sub, multi, division.
  - Multi is shift and add
  - Div is shift and sub
  - Add is a little more than xor
- Computers don't do absolute value
- What if series converge slowly, then do Algebra, rewrite equation to write faster code.
- Floating arthmetic.

Normal Math

Normal #'s exact.

Comp Math

Numbers are approximate

- Don't compare Floating point numbers using equality.

Lec 6

What is array?
- homogenous collection of element
- Array here 1 dimension called vectors

Arrays are order.
start at 0, end at a[n-1]

a[0] ← base address of the array.

(does not care if you go off the array.
a[10] = a[10+1]

Arrays → vectors → matrices.

$m = \begin{bmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \\ 1 & 2 & 3 \end{bmatrix}$ m[3][3];

In C everything is "call by value"

but arrays used pointers.

"&" (address of) operator.

sizeof operator tells us the number of bytes use in variable.

Works on array, structures.

CSE 13x

Lec 7

- Bad algorithms run slow
  Good ones runs fast.

Common function	10	30
$\log(x)$	2, 3	3, 4
x	10	30
$x \log(x)$	~~2.3~~	102
x^2	100	900
x^3	1000	27000
2^x	1024	1. Bill
$x!$	3 mil	10 Gag

Ex

$O(\log(n))$	Binary
$O(n)$	Find min
$O(n\log(n))$	Merge
$O(n^2)$	Bubble
$O(n^3)$	Matrix Mult
$O(2^n)$	Enumerat subset
$O(n!)$	Enumerate Permutation

For loops = $O(n)$

2 nested for loops = $O(n^2)$

Time is "how many steps it takes"

Space is "how much memory it uses"

CSE 13S

Lec 8

Sorting

- Sorting - Keep things in a well defined order.
- There are partial and total ordering.

Bogo sort

- enumerate all possible orderings of n objects
- Pick the one that is in order.

$O(n^2)$

Insertion and Selection

Bubble sort Quick (worst case)

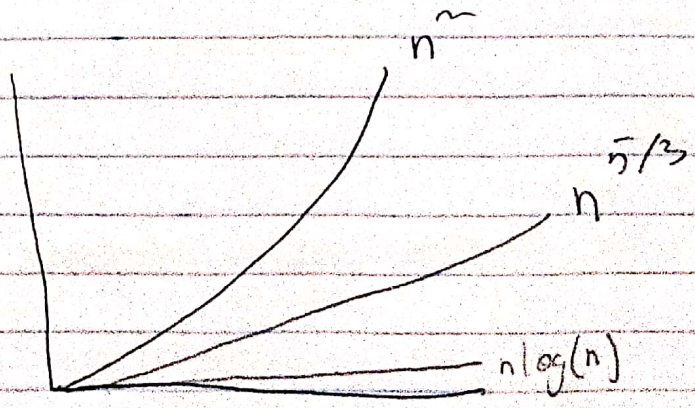
Shell sort is an $O(n^{5/3})$

$O(n \log(n))$

Merge

heap

Quick (avg case)



Heaps

Single node is a heap.

If a heap if parent is a heap and trees rooted at both children are heaps.

A parent's value is greater that child.

No order among children.

CSE 13s

Lec 9

- Pointer - variable that holds a memory address
  Pointer that don't contain address are set to NULL
  NULL = 0
- Memory stored in registers that can be access by address.
- Can assign a pointer the address of a variable using (&)
- Multiple pointers can point to same address
- Pointer can be dereferenced by (*)

"++" increment to next address (increment by 4 bytes assuming 32-bit int)

"--" decrement to previous address.

"+" pointer + numeric value. (no pointer pointer)

"-" subtract 2 pointer

Declaring array in a functions allocates it on a stack.

CSE 13s

Lec 10

Dynamic Memory Allocation

- Allocating memory for variables on the heap during run time.

Variables on stack don't last beyond current scope.
Variables dynamically allocated on the heap can be access beyond scope

- 3 functions to allocate memory.
  - malloc
  - calloc
  - realloc

Alocated memory must be freed.

Heap is slow because it need pointers

Malloc don't check if overflow.

Calloc

- void * calloc (size_t nmemb, size_t size).

Returns * to nmemb.

- Clears memory.

Realloc

- If enough space to make bigger, Done fast.

Memory leaks occur if memory is not freed.

Create → use → delete

CSE 13s

Lec 11

A recursive function is defined in terms of itself.

Ex. $\Sigma k = k + \Sigma(k-1)$

Peel things off stack after function call.

List of slow
- Hard drive
- SSD
- D-RAM

Fast

- Use recursion when it makes sense
- Binary Seach is a fast as we can go.

$O\log(n)$

Times you can divide by 2 = $\log(n)$

Computers don't do recursion.

Lec 9

Graph defined as $G = \langle V, E \rangle$

vertices, edges.

Vertices

$V = \{v_1, v_2, \ldots v_n\}$

Edges

$E = \{\langle v_i, v_j \rangle, \langle v_p, v_q \rangle \ldots\}$

Edges may have direction (directed)
or no direction (undirected)

2 ways to search in Graph

Breadth-first search
uses queue
Explore set of verticies that are immediatly reachabl

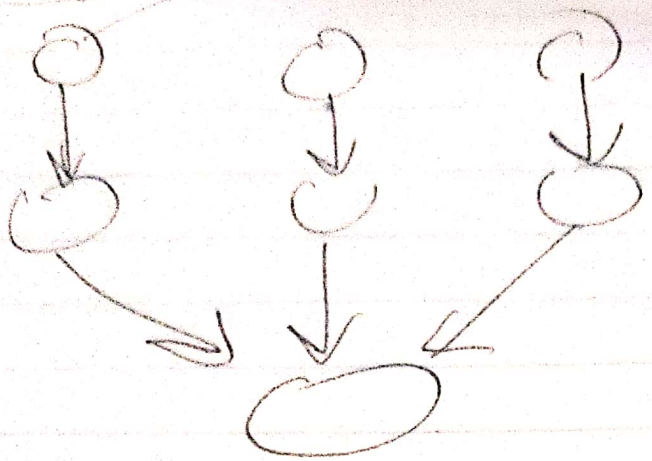
Depth First search

Starts at the bottom, uses recursion.

DAG (Directed Acyclic Graph)

Tree is a DAG

Topological Sort



Hamiltonian Path

Visits vertices exactly once.

Eulerian Path

Each edge once.

Lec 13

CSE 135

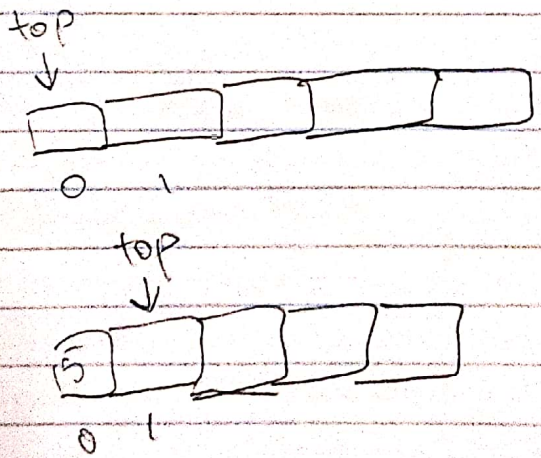
Linked List

Start at begining follow to next

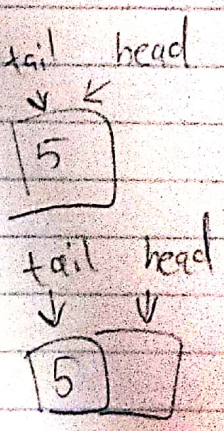
Sequential order.

Data type may be placed in a stack or Queue

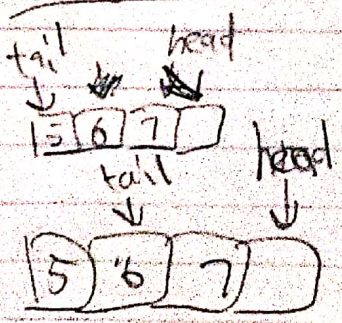
Push items to stack.



Enque:



Dequeue



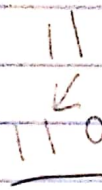
CSE 13S

Lecture 14

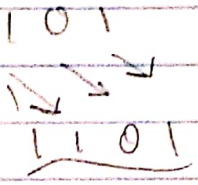
Bit	1	0/1
Nibble	4	Hex
Byte	8	ASCII Smallest addressable on comp.
Half Word	16	
Word	32	
Long Word	64	

Arithmetic

Shift Left



Shift Right



How to Set a bit

Shift and or

Clear bit

Bit want

move to position

flip entire thing

Getting bit

Shif

And

Shift.

CSE 135

- D-ram does not survive being turned off.
- Computers has millions of files.

Files has base names then extensions

"File.bak"
"File.c" - Mostly true in Windows.

- Magic Number - we make up and the computer tries to figure out what to do.
- Unix files: Sequence of bytes.

Executable

Header
- Magic Number
- Text size
- Data size
- BSS size
- Symbol table
- Entry table
- Flags

Text

Data

Relocation bits
Symbol table

File Operators

Create	Append
Delete	Seek
Open	Get Attributes
Close	Set Attributes
Read	Rename
Write	

Directories

Naming is better than using numbers.

Directory Operators

Create	Readdir
Delete	Rename
Opendir	Link
Closedir	Unlink

i-nodes : a struct
1st few entries
represents sectors on disk

next are pointers to sectors on disk

Linked lists

Linked List

- No fixed Memory Allocation.
- No need to shift elements.
- Only update the address to next pointed

- storing pointer to next node requires extra memory.

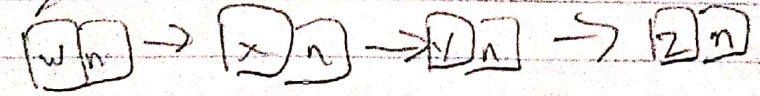
Caches small memory faster than D-ram.

L1 cache → close to CPU speed

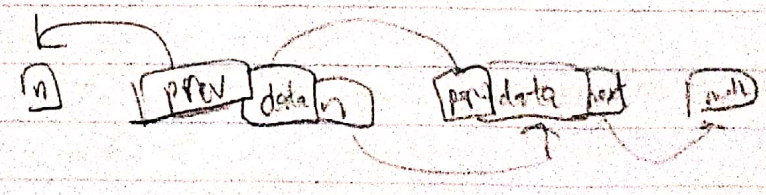
L2 cache

L3 cache

Circularly Linked list



Double linked list



CSE 13S

Trees.

- A tree is a type of directed acyclic graph (DAG) usually composed of Nodes.
- A Binary tree has up to 2 children
- Root, starting point of the tree. If NULL, then empty.
- child: a node connected to a parent.
- leaf: no children nodes

goes with stack

Preorder
- if root
- print
- go left
- go right

Inorder
- if root
- go left
- print
- go right

sorts the ~~code~~ tree

Postorder
- left
- right
- print

Level in a tree
- use queue
- BFS
- root in queue
- children queue
- } repeat.

Smallest
Follow left nodes

Largest
Follow right links

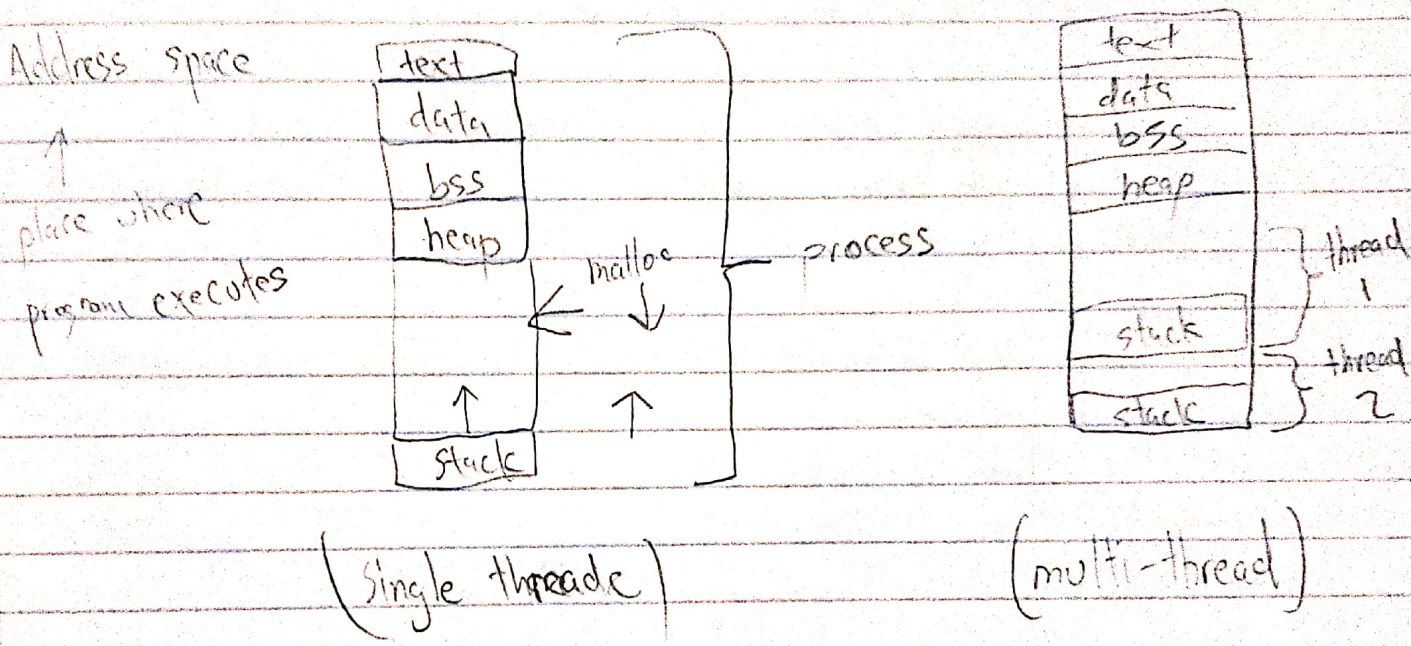
Balanced tree
$O(\log(n))$

Unbalanced
$O(n)$

Multithreading

Only one process can run on a single CPU core at a time.
- Multi core can support multiple processes.

A process is code, data, stack



Why Use Threads?

Faster to create and destroy.
Allow single application to do many things.

#include pthread.h

pthread_create create new.
pthread_join wait for a thread to be complete.

When a thread gets made
its own id.
pointers.
Race condition: threads happening same time

Critical regions to provide mutual exclusions
to fix race conditions

4 conditions for mutual exclusions
No two processes may simultaneously be in critical region

No assumptions may be made about speed. /# of CPUs

No process running outside critical region.
and blocks processes

Deadlock: process or thread stopped b/c reasource needed is held up by another processes.
Locking out threads from critical region is done using mutex.
pthread_mutex_lock
pthread_mutex_unlock.

Ostrich algorithm
pretend no there.

Recovering from Deadlocks.

Preemption
rollback: copy things and roll backs.
killing process

preventing Deadlock
Try to attack
mutual exclusion
Hold & wait
No preemption
Circular wait