CSC 225 Final

Conner Leverett Tue Dec 15 14:14:25 PST 2015

What is an algorithm

Sequence of unambiguous instructions for solving problems and obtaining the desired output for any legitimate input in a finite amount of time

---

Big oh formula

There exists a c and Nnot such that f(n) <+ constant * some g(n) for all n greater than or equal to Nnot

---

Names and order of Big OH

Constant O(1)
Log O(logn)
Linear O(n)
Linearithmic O(n*logn)
Quadratic O(n^2)
Polynomial O(n^k)
Exponential O(2^n)

What is the best sorting algorithm running time?

O(n logn)

Put these in order for fastest to slowest square root n, logn, n logn, n^2, N^k, logn^2, nlogn, 2^n

logn, logn^2, square root n, n, nlogn, n^2,n^3, 2^n

Big Oh

f(n) = Og(n) iff there exists a constant c>0 and Nnot such that f(n) <= c *g(n) for every n >= Nnot

---

What is the running time of summation formula?

n(n+1) all over 2

Square summation formula?

n(n+1)(2n+1) all over 6

Cube summation formula?

(((n(n+1))/2)^2

Big Omega

f(n) is big omega of g(n) iff g(n) is big oh of f(n)

Big means __
Big Omega Means __

at most
at least

If f(n) = O(g(n)) and f(n) = Omega(g(n))

Then f(n) is big theta of g(n)

Induction steps

Basis Step - Check to make sure works for n=1 or n=2 , just the lowest number
Induction Hypothesis - Assume true for n=k
Induction step - show true for n = k+1

Stack operations and running time. What kind of operator is it?

Push - puts object at top O(1)

Pop - Remove from stack and return O(1)
Peek - What's on top O(1)

Last in first out

Pros and cons if array based stack

con- must assume fixed upper bound
pro - simple and if have good estimate of how many values is a powerful tool

---

Queues? Operations and running times

FIFO
enqueue - O(1)
dequeue - O(1)

What does an ADT specify

data storage
operations on data
error conditions

Usage of stack?

infix to postfix

Total order vs partial order

total = 1,2,3
partial = {csc 100, csc 115, csc 225},{Seng 200, Seng 202}

Sorting Algorithms and times

Insertion, Bubble Selection all N^2
Merge, Quick, Heap all Nlogn (quick can have worst of n^2)
Radix n = restricted input

How does selection sort work?

First pass finds smallest, swap with position 0, finds next smallest switches with position 1 and so on

---

Insertion Sort

Inserting not swapping, makes sure element 1 is in order, then element 1 and 2, then element 1,2 and 3, and so forthm

---

Do a merge sort!

What is a stable sort?

If it guarantees that the original order of equal elements is preserved

---

What is an inplace algorithm?

One that can do work in array - eg insertion sort requires O(1) extra storage space

---

Can quick sort perform better than nlogn?

Yes but worst n squared but almost never this

Worst case recurrence and best case of QS

T(n-1)+2n
2T(n/2)+2n

QS linear running time?

O(n)
T(n/2) +n

How to check if good pivot?

Compare sizes of lower and greater

What's faster normal or random quick sort?

Random

What is the level of the root of a binar tree?

0

At each level of a binary tree there are at most __ nodes

2^L
Eg 0 -1
1--2
2--4
3--8

Root of tree is at depth __ , depth is depth __ of parent

0
+1

Number of leaves in a binary tree is atleast?

height +1

---

the number of internal nodes in a binary tree is at least __ and at most __

h
2^h-1

Number of leaves in any binary tree of height h

greater than h+1 but less than 2^h

Number of internal nodes in any binary tree of height h

Greater than or equal to h but less than or equal to (2^h)-1

Number of internal nodes is __ __ than leaves

1 less

In place sorting algorithms

Heap, insertion, selection, bubble

Stable Sorting algorithms

Everything but heap and selection

Height? Depth?

Bottom up, eg leaves at height 0

Goes down, eg root at Depth 0

Preorder?

Depth Search
Takes O(n) time
Root visited first

Post Order Traversal? What is this useful for?

Visit child then root
therefore top root is the last node visited

When want to go bottom up

In order Traversal
What is this useful for?

Nodes visited left to right

useful in expression trees

What is depth the same as?

Also known as level

Time complexity of preoder, inorder, and postorder?

O(n)

How to number nodes?

Root is 1 left node is 2x parent and right node is (2Xparent) +1

Priority Queue?

Container of elements each having an associated key
Keys determine priority used to remove elements

Operations of Priority Queue?

Insert
Delete min or delete max (just one)

Heap is a realization of

a priority queue

Heap order property

Every node V, other than the root, its key stored is smaller than or equal to the key stored at V's parent.

Shape property

A binary tree with height h is complete if all levels have max # of nodes possible
and
at level h the internal nodes are all to the left of the leaves

An example of a complete binary tree?

a heap!

Running time for heap insert? Running time for heap remove?

Insert - O(h) where h is the height of the tree
Remove - O(h) where he is the height of the tree (which should be logn in both cases)

Running times of bubble up and bubble down operations in a binary tree?

logn

How does a bucket or bin sort work?

partitions a set of elements into a finite number of buckets
each bucket is then sorted

What is the fastest sort?

nlogn for comparison sort
but as bucket not comparison based can run faster than this

What does bucket sort run in?

O(n+N)
n=number of items
N= number of buckets

MSD and LSD radix sort?

most significant digit
least significant digit

Time complexity of radix sort?

O(d(n+N)) where d =number of digits

How many leaves must a decision tree have?

at least n! leaves as must have every permutation of ordering

eg 8 numbers would have 8! leaves

Read induction on trees!

DO ITT

In order traversal will return...

sorted order! Goes left to right

If you're searching for something in a tree how do you know if it's not there

When you've hit a leaf

Insert Delete and Find runnig times of BST?

O(h) where h is the height of the tree

How does insert work for a BST?

Run find, if it doesn't exist put it into empty node
If number alreayd exists find it then go to the **left** of it and insert there

---

How does delete work in BST?

if has one child, it's child takes it's place
If has two children go to the left node then as far right as possible **or** go right then as far left as possible

---

What is the height of a BST?

typically log n but could be O(n) if all nodes inserted were greater than and then is just a glorified linked list

---

AVL tree?

BST with height balance proportionally rule

What is the height balance proportionally rule?

Every internal node's children's height can differ by at most 1

The height of an AVL tree?

logn

Worst case running time of hashing by chaining?

O(n) - all values go to the same spot

If we have a good hash value what will happen?

will distribute the values equally

What is the load factor?

the number of items over the number of slots - we want this to be small

---

Average case search time __ as load factor __

increases
increase

Average case search time of open addressing?

1/1-load time
where load time is number of items/number of slots

What is a graph?

Set of vertices and a collection of edges

What is a vertices degree?

How many neighbours you have

Types of degree of a directed graph?

two types in degree and out degree

What is a path in a graph?

sequence of vertices such that there are edges from vn to vk

What is a connected graph?

If every pair of vertices is connected, eg can get from one node to any other node

---

Different types of trees?

Rooted trees, unrooted trees, binary trees, AVL trees, heaps

What is a simple graph?

no self loops and no parallel or multi edges

What is a multiple edge in a graph

vertices a and b are connected more than once

A complete graph?

A simple graph where an edge connects every pair of vertices

What is a subgraph?

where G'=V',E' where V' is a subset of V and E' consists of edges (v,w) in E such that both V and W are in V'

What is a tree Graph?

there is no way to walk down the path and get back to the original node

is connected
has no cycles

What is a forest?

collection of trees

Number of edges in a tree?

n-1

What is a spanning tree?

-spanning subgraph that is a tree
-not unique unless the graph is a tree
-have applications in communication networks

A simple path?

if all vertices on the directed path, except for possibly the first and last are distinct, eg no parallel or multi edges

---

What is the length of a directed path?

number of arcs in the path

Space, add edge, check if V is adjacent to W, and output all vertices adjacent to V for Edgelist

|E|, O(1), |E|, |E|

Space, add edge, check if V is adjacent to W, and output all vertices adjacent to V for Adjacency Matrix

|V|^2, O(1), O(1), |V| - as scan through vertices

Space, add edge, check if V is adjacent to W, and output all vertices adjacent to V for Adjacency List

O(|V|+|E|), O(1), (Size of linked list), (Size of linked List)

Most of the time this is the best method

There are often much more __ than __ in graph

N^2 edges only when __

edges, vertices
complete graph

What is an edgelist?

Just the edges listed together eg
(1,5)
(2,3,10)
(4,6)

Adjacency Matrix?

A matrix representation of adjacency

Adjacency list

A linked list representation of what each vertices is connected to

---

DFS?

-start at a fixed vertex v and mark v as visited
-from v select an arbitrary edge not already explored
-if u has not been visited yet we proceed from that node recursively in a depth first manner,
-otherwise backtrack to vertext v and explore next incident edge

---

Discovery edges in a DFS form a __

tree

Number of discovery edges in a DFS?

n-1

In a DFS tree siblings will __

never be connected

How to tell if two nodes are connected with DFS?

start DFS at nodex if nodey is visited then they are connected

---

Does DFS or BFS give you shortest path?

BFS

Running time of DFS?

O(|E|+|V|)

What is 2 connected?

can remove any link and graph is still connected

What is a DAG?

Directed Acyclic Graph
Graph with no cycles
All edges are forward
More general than trees but less general than arbritary graphs

---

What is a labeled graph?

-a graph with edge labels
-denote edge attributes

No vertice connected to vertice at level __ in a tree

n+2
This is because if we suppose there is an edge which jumps level then that node should actually be at a lesser level

A graph is bipartite if

v can be partitioned into sets U & W such that all edges in G have one end point in U and one end point in W

If there is no edge between two vertices with same level number then..

we can conclude graph is bipartite

If there is atleast one edgde between two values of same level then we can conclude..

graph is not bipartite

Properties of DFS

-Visits all edges and vertices in the connected component of a node
-Discovery edges create a spanning tree of the connected component of V

Strongly Connected Components

two vertices u and v are strongly connected if there is a (directed) path from u to v and a there is a (directed) path from v to u

For a BFS if there is an edge within same level what does that mean?

There is an odd cycle in the graph

If there is an odd cycle in G what does that mean?

No bipartite

DFS in directed graphs?

same but now can only go where direction allows

Preorder? Postorder?

Increasing arrival time
Decreasing departure time

Forward (u-->v)

a=arrival
d=departure
a(u)<a(v)

Back (u-->v)

a=arrival
d=departure
a(u)>a(v)
d(u)<d(v)

Cross (u-->v)

a=arrival
d=departure
a(u)>a(v)
d(u)>d(v)

Tree (u-->v)

a=arrival
d=departure
a(u)<a(v)
d(u)>d(v)

Given a digraph, does it have a single cycle?

Case 1) run DFS- if back edge is found then G has a cycle
Case 2) no back edges are found, can't create cycles through cross and forward edges

What happens when you reverse post order?

should only have forward edges

If there is just one directed graph then. If there are multiple?

it is connected
then unconnected

What does it mean for a digraph to be strongly connected?

if every pair of vertices in G are strongly connected

One method for checking if a graph is SC?

Run DFS at every vertex and check if all vertices are visited
This is too slow though

Fastest method for checking if digraph is SC?

Run DFS(node) check to make sure it visits all vertices
Compute G' (G reverse) run dfs on the same node and check if all vertices in G' are visited
O(|E|+|V|)

In any DAG there are two values

sink - only incoming
source - only outgoing

Every run of DFS started at sink component gives

the strongly connected components

How to identify sink components?

Run DFS on Greverse starting at) and compare the postorder listing of vertices

---

Transitive closure

The transitive closure G of a directed graph contains edges between all vertices reachable by a directed path of any length

---

Transitive Closure G* of G is a digraph such that

1) The vertices in G* are the same as the vertices in G
2) G* star has an edge (u,v) whenever u reaches v in G

What are the 4 basic algorithms which can construct G*?

DFS or BFS
Identity SCC
Warshals
Matrix Multiplication

Time complexity of transitive closure?

O(n^3)
But with matrix multiplcation somewhere between n^2 and 2.3

MST?

Minimum Spanning tree
-Subgraph of undirected weighted graph
-Covers all vertices and has |V|-1 edges
-Total cost associated with three edges is the minimum among all possible spannings trees
-not necissarily unique

Weight of a spanning tree?

Sum of all edges

What is a greedy algorithm?

A global optimal can be reached by a series of locally optimal choices

---

Can you solve MST by brute force?

No

How to create an MST from a spanning tree?

Create a cycle, then implement edge and throw away biggest weight
Do this repeatedly

Proof of why graph is still connected when add cycle then throw away edge

Case 1) if path between u and v not thrown out then G is still connected
Case 2) If add edge to spanning tree which creates a ccle, if you delete an edge of this cycle graph is still a spanning tree

---

Kruskal Time Complexity

Constructs a MST for a connected weighted graph G with n vertices and m edges in O(mlogn) time

---

If all edge weights in a tree are distinct then __. If they aren't unique then __.

MST is unique
MST is not unique

Kruskal's Proof of correctness

look at it

Given any cut of an edge weighted graph...

the edge of minimum weight belonging to the cut is a part of the MST

Cut

A set of edges such that if these edges are removed the graph becomes 2

Prim's Algorithm

Apply cut property many times
