CS 385 Test 2 Review

4/13/16

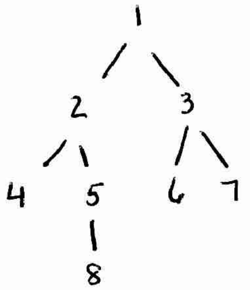
Graphs

	Matrix	List
BFS	$\Theta(\|V^2\|)$	$\Theta(\|V\|+\|E\|)$
DFS	$\Theta(\|V^2\|)$	$\Theta(\|V\|+\|E\|)$

Matrix $\begin{array}{c|cc} & 1 & 2 \\ \hline 1 & 0 & 1 \\ 2 & 1 & 0 \end{array}$

List
1 → 2
2 → 1

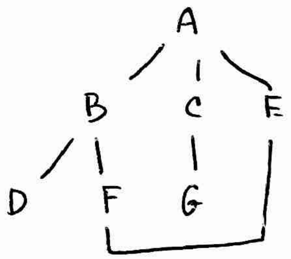
BFS



1, 2, 3, 4, 5, 6, 7, 8

Starts at root.
Explores all neighbor nodes before moving to the next level.

DFS

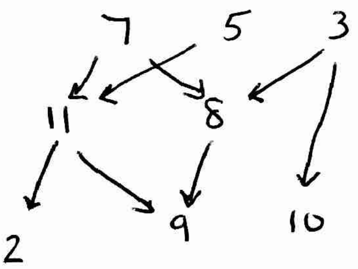


A, B, D, F, E, C, G

Starts at root
Goes as far down as possible before retracing its steps

Topological Sort

$\Theta(|V| + |E|)$



Array	2	3	5	7	8	9	10	11
In Degree	~~1~~ ~~0~~	~~0~~	~~0~~	~~0~~	~~2~~ ~~1~~ ~~0~~	~~2~~ ~~1~~ ∅	~~1~~ ~~0~~	~~2~~ ~~1~~ ~~0~~

~~Order: 2, 3, 5, 7~~

Order : 3, 5, 7, 8, 11, 2, 9, 10

Use A priority Queue. A node is complete when it's in degree is 0.

Lomuto Partition

Partitions an array around a pivot

```
Lomuto Partition(A[l...r])
 p ← A[l]
 s ← l
 for i ← l+1 to r do:
     if A[i] < p
         s ← s+1;
         Swap(A[s], A[i])
Swap(A[l], A[s])
return s
```

Quick Sort

```
Quick Sort (A[l...r])
    if l < r
        s ← partition(A[l..r])
        QuickSort(A[l..s-1])
        Quick Sort(A[s+1..r])
```

$$2 \cdot T(n/2) + \overset{\text{Partition}}{\Theta(n)}$$

so $a = 2$

$b = 2$

$d = 1$

$2 = 2$

$\Theta(n \log n)$

Merge Sort

```
MergeSort(A, lo, hi)
    if lo<hi
        mid = (lo+hi) / 2
        mergesort(A, lo, mid)
        mergesort(A, mid+1, hi)
        L = lo
        H = mid+1
        for k ← lo + hi
            do if L<=mid and (H>hi or A[L] <= A[H])
                then scratch[k] = A[L]
                     L += 1
                else scratch[k] = A[H]
                     H += 1
```

```
for k ← lo to hi
    A[k] = scratch[k]
```

$\Theta(n \log n)$

Master Theorem

$T(n) = a \cdot T(n/b) + f(n)$ where $f(n) \in O(n^d)$ $d \geq 0$

if $a < b^d$ $T(n) \in \theta(n^d)$

$a = b^d$ $T(n) \in \theta(n^d \log n)$

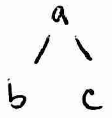
$a > b^d$ $T(n) \in \theta(n^{\log_b a})$

a = # of recursive calls made

b = how much that n is broken down in recursive call

f(n) = complexity of the rest of the code

Binary Search Tree



$a > b$

$a \leq c$

Internal Node → Node with children

Leaf → Node w/ no children

Max Width → ~~Max filled~~ # of nodes in most filled level

Height → Longest # of edges from root to leaf

Pre Order	Root L Rt
InOrder	L Root Rt
Post Order	L Rt Root

Russian Peasant Multiplication

$n \cdot m = \frac{n}{2} \cdot 2m$

if n is even

if n is odd $\frac{n-1}{2} \cdot 2m + m$

$\underline{n}$ $\underline{m}$

$15 \cdot 27 + 27$

$7 \cdot 54 + 54$ 405

$3 \cdot 108 + 108$

$1 \cdot 216 + 216$

405

Binary Reflected Gray Code

```
BRGC(n)
    if n=1 make List L containing bit strings 0 and 1 in this order
    else generate list L1 of bit strings of size n-1 by BGRC(n-1)
         copy list L1 to L2 in reversed order
         add 0 in front of each bit string in L1
         add 1 in front of each bit string in L2
         append L2 to L1 to get list L
    return L
```