

Agenda

- 1) Introduction
- 2) Big-Oh
- 3) DFS example
- 4) Odd Cycle

1] TA: Evan Ettinger

OH: M 2-4pm Ebu3b B210

e-mail: eettinger@cs.ucsd.edu

*Note: please e-mail personal issues only, HW or test questions should go on the webboard.

2] Big-Oh

- Used to analyze running time and memory usage of an alg.

(time complexity)

(space complexity)

E.g. Algorithm takes input of size n

A: runs in 2^n time

B: in n^2 time

n	10	100	1000	10000
A	2	10^{24}	$\approx 10^{301}$	$\approx 10^{3000}$
B	1	100	10^4	10^8

*Note: $\approx 10^{80}$ atoms in observable universe.

We say Alg. B is more efficient than A since as n gets large B will always run faster than A.

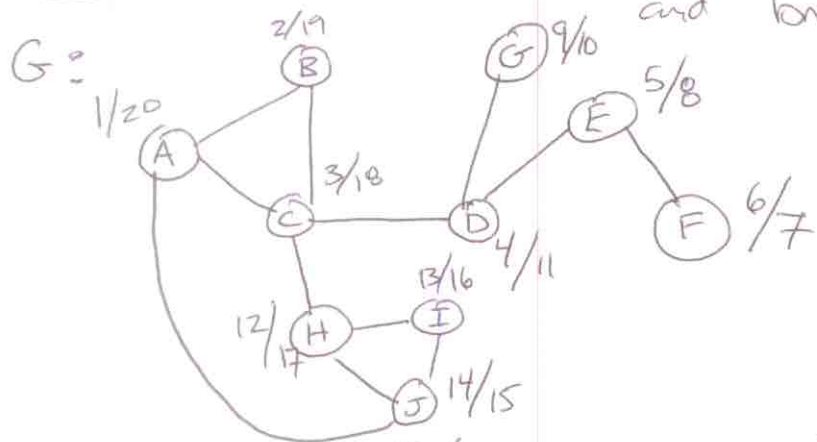
- Defn: Let $f(n), g(n)$ be ~~pos. reals~~ functions from positive ints to pos. reals. We say $f = O(g)$ if we can find a constant $c > 0$ s.t. $f(n) \leq c g(n) \forall n > 0$.
 Also, $f = \Omega(g) \Leftrightarrow g = O(f)$ and $f = \Theta(g) \Leftrightarrow f = O(g)$ and $f = \Omega(g)$

[3] DFS example

[3]

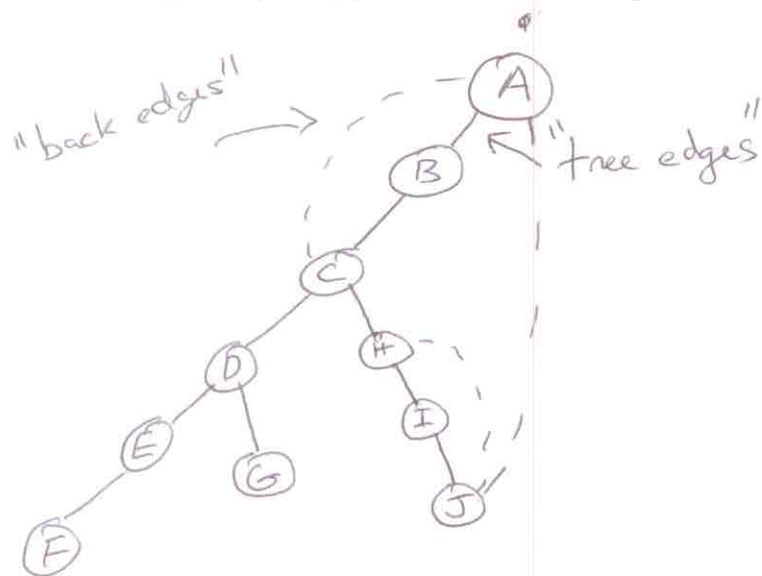
- Depth First Search is an algorithm that explores the structure of a graph.

Ex: Run DFS on the following graph starting @ A and breaking ties alphabetically



• Note: x/y means node had pre # x and post # y

- Here is the corresponding DFS-tree of G



4 Odd Cycle

4

Devise an algorithm as follows (linear time):

Input: Undirected $G=(V,E)$

Output: Does G have an odd cycle? (Yes/No)

• How can we use DFS to solve this?

• Idea: Undirected G has only tree/back edges and back edges are cycles in G !

~~paths of nodes on sides of back edge to see if cycle is even/odd~~

• Alg

OddCycle(G)

1) = Run DFS(G), coloring each child the opposite color of its parent (R, B edges)

2) = For each back edge $e=(u,v)$ in the DFS-tree:
if (color(u) == color(v))
output "Yes" and halt

3) = output "No"

• Run-time analysis: DFS takes $O(|V|+|E|)$, 2) is at most $O(|E|)$
So, OddCycle is $O(|V|+|E|)$.

• Correctness: Claim: G has an odd cycle in it iff OddCycle(G) outputs "Yes".

1) " $\Rightarrow$ " Assume G has an odd cycle. Let (u,v) be the backedge for this cycle in G . Since all other edges in the cycle are tree edges, colors alternate along the cycle causing color(u) == color(v).
2) " $\Leftarrow$ " Assume OddCycle(G) outputs "Yes" then a backedge has the same color. So the path from $u \rightarrow v$ using tree edges is even + one edge (u,v) is odd cycle.

Ex: Is $f = O(g)$? $f = \Omega(g)$? $f = \Theta(g)$? [2]

(a) $\frac{f}{g} = \frac{n^{1/3}}{n^{2/3}}$

• Can we find $c > 0$ s.t. $n^{1/3} \leq c \cdot n^{2/3} \forall n > 0$?

$$\frac{n^{1/3}}{n^{2/3}} = n^{-1/3} = \frac{1}{n^{1/3}} \leq 1 \quad \forall n \text{ integer } > 0$$

so $c=1$ shows $f = O(g)$.

• Can we find $c > 0$ s.t. $n^{2/3} \leq c \cdot n^{1/3}$?

$$\frac{n^{2/3}}{n^{1/3}} = n^{1/3} \not\leq c$$

For any c we choose we can find ~~some~~ an n_0 s.t.
 $n_0^{1/3} > c$.

e.g. $c=10$

$n_0=1001$ will do the trick.

In general

$n_0 = \lceil c^3 \rceil + 1$ does the trick

So, $f \neq \Omega(g)$

$$\boxed{f = O(g)}$$

(b) $\frac{f}{g} = \frac{\log n}{n^{0.01}}$

$\frac{g}{n^{0.01}}$

• $\log n \leq c \cdot n^{0.01}$?

so $f = O(g)$

L'Hopital's rule

$$\frac{\log n}{n^{0.01}} \stackrel{\text{L'Hopital's rule}}{=} \frac{\frac{1}{n}}{0.01 n^{-0.99}} \leq 100 \quad \forall n > 0 \text{ ints.}$$

" $\frac{1}{0.01 n^{1.99}}$

• $n^{0.01} \geq c \cdot \log n$? No, for similar reasoning.

Note: In your hw you can use rules 1-4 on pg 8 as justification ~~and~~ to your answers.