Discussion Week 1

Agenda

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

1) TA: Evan Ettinger
   OH: M 2-4pm Ebu3b B210
   e-mail: ettinger@cs.ucsd.edu
   *Note: please e-mail personal issues only, HW or test questions should go on the blackboard.

2) Big-Oh

- Used to analyze running time and memory usage of an algorithm (time complexity, space complexity)

  E.g., Algorithm takes input of size n
  A: runs in $2^n$ time
  B: in $n^2$ time

<table>
<thead>
<tr>
<th></th>
<th></th>
<th>10</th>
<th>100</th>
<th>1000</th>
<th>10000</th>
<th>100000</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>2</td>
<td>$10^2$</td>
<td>$10^3$</td>
<td>$10^4$</td>
<td>$10^5$</td>
<td>$10^6$</td>
</tr>
<tr>
<td>B</td>
<td>1</td>
<td>$10^0$</td>
<td>$10^1$</td>
<td>$10^2$</td>
<td>$10^3$</td>
<td></td>
</tr>
</tbody>
</table>

- 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 functions from positive ints to pos. reals. We say $f = O(g)$ if we can find a constant $C > 0$ s.t. $f(n) \leq Cg(n)$ for all $n > 0$.

  Also, $f = \Omega(g) \iff g = O(f)$ and $f = \Theta(g)$.
DFS example

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

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

G:

```
A -+ B
  |
  v
C -+ D
    |
    v
E -+ F
    |
    v
H -+ I
  |
  v
J -+ K
```

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

- Here is the corresponding DFS tree of G:

```
  A
  |
  v
B -+ C
  |
  v
D -+ E
  |
  v
F
```

"back edges"  "tree edges"
Odd Cycle

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 $G$!

Algorithm:

$\text{OddCycle}(G)$

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

2) For each back edge $e=(u,v)$ in the DFS tree $G$
   - 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

$\text{OddCycle}(G)$ outputs "Yes",

1) $\Rightarrow$ Assume $G$ has an odd cycle. Let $(uv)$ be the back edge for this cycle in $G$. Since all the edges in the cycle are tree edges, colors alternate along the cycle causing color($u$) == color($v$).

2) $\Leftarrow$ Assume $\text{OddCycle}(G)$ outputs "Yes" then a back edge has the same color, so the path from $u$ to $v$ using tree edges is even and one edge $(uv)$ is odd cycle.
Example Is \( f = \Theta(g) \) if \( f = \Omega(g) \) ?

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

- Can we find \( c > 0 \) such that \( n^{1/3} \leq c - n^{2/3} \) \( \forall n > 0 \)?

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

So \( c = 1 \) shows \( f = \Theta(g) \).

- Can we find \( c > 0 \) such that \( n^{2/3} \leq c - n^{1/3} \)?

\[
\frac{n^{2/3}}{n^{1/3}} = n^{1/3} \nless c \quad \mathbf{\text{no}}
\]

For any \( c \) we choose we can find \( \exists \) \( c_{0} \) such that \( \exists n_{0} \) \( n_{0}^{1/3} > c \).

E.g., \( c = 10 \)

\( n_{0} = 1001 \) will do the trick.

In general, \( n_{0} = \lceil c^{3} \rceil + 1 \) does the trick.

So, \( f \neq \Theta(g) \)

\[
\boxed{f = \Theta(g)}
\]

(b) \[
\frac{f}{\log n}
\]

- \( \log n \equiv c - n^{0.01} \) ?

\[
\frac{\log n}{n^{0.01}} = \frac{\log \frac{1}{n^{0.01}}}{0.01 n^{1.99}} \leq 100 \quad \forall n > 0 \text{ int},
\]

so \( f = \Theta(g) \)

- \( n^{0.1} \geq c \cdot \log n \) ? \( \mathbf{\text{No}} \), for similar reasoning.

Note: In your hw you can use rules 1-7 on pg 8 as justification to your answers.