CS157 HW1 Seung Ji Lee (sl136)

1/

See hw_template.

2/

1) $n! = n(n-1)(n-2)\cdots(1) \le n(n)(n-2)(n-3)\cdots 1 \le n(n)(n)(n-3)\cdots 1 \le n(n)(n)\cdots(n)$
$\le n^n.$

2) $n! = n(n-1)(n-2)\cdots(\lceil \frac{n}{2} \rceil)\cdots 1 \ge (\lceil \frac{n}{2} \rceil)(\lceil \frac{n}{2} \rceil)\cdots(\lceil \frac{n}{2} \rceil)(\lceil \frac{n}{2} \rceil-1)(\lceil \frac{n}{2} \rceil-2)\cdots 1$
$\ge \lceil \frac{n}{2} \rceil^{\lceil \frac{n}{2} \rceil}(\lceil \frac{n}{2} \rceil-1)! \ge \lceil \frac{n}{2} \rceil^{\lceil \frac{n}{2} \rceil}$

3) False, since $n^n \ge n!$, $\nexists$ no c, constant such that
$c|\log n^n| \le |\log(n!)|$ $\forall n$.

4) False, since $n^n \ge n!$, $\nexists$ no constant c such that
$c|n^n| \le |n!|$ $\forall n$.

3

1) $T(n) = 2T(\frac{2n}{3})$. $T(n) = 1$ when $n \le 1$.

$$T(n) = 2T\left(\frac{2n}{3}\right) = 2\left(2T\left(\frac{2^2}{3^2}n\right)\right) = 2^k T\left(\left(\frac{2}{3}\right)^k n\right) = 2^k$$

where $\left(\frac{2}{3}\right)^k n \le 1$ but $\left(\frac{2}{3}\right)^{k-1} n > 1$.

Then $n \le \left(\frac{3}{2}\right)^k$, $n > \left(\frac{3}{2}\right)^{k-1}$.

Then, $k \ge \log_{\frac{3}{2}} n$, $k < \log_{\frac{3}{2}} n + 1$

Then $k = \lceil \log_{\frac{3}{2}} n \rceil$

Then $T(n) = 2^{\left\lceil \frac{\log n}{\log(\frac{3}{2})} \right\rceil} = \Theta(2^{\log_{\frac{3}{2}} n}) = \Theta(n)$

2) $T(n) = T(2n/3) + 2$ $T(n) = 1$ when $n \le 1$.

$$T(n) = T\left(\frac{2n}{3}\right) + 2 = \left(T\left(\frac{2^2}{3^2}n\right) + 2\right) + 2 = T\left(\left(\frac{2}{3}\right)^k n\right) + 2k = 2k + 1$$

where $\left(\frac{2}{3}\right)^k n \le 1$ but $\left(\frac{2}{3}\right)^{k} n > 1$

By similar calculation as 3.1,

$k = \lceil \log_{\frac{3}{2}} n \rceil$

Then, $T(n) = 2(\lceil \log_{\frac{3}{2}} n \rceil) + 1 = \Theta(\log n)$

3) $T(n) = 3T(n-5) + n \qquad T(n) = O(1)$ when $n \le 5$

$$T(n) = 3T(n-5) + n = 3(3T(n-10) + n - 5) + n$$
$$= 3^k T(n-5k) + kn - (1+2+\cdots+k-1)5$$
$$= 3^k O(1) + kn - \frac{(k-1)(k)}{2} 5.$$

where $n - 5k \le 5$ but $n - 5(k-1) > 5$.

Then $\frac{n-5}{5} \le k < \frac{n-5}{5} + 1$

$\therefore k = \lceil \frac{n-5}{5} \rceil$

Then $T(n) = 3^{\lceil \frac{n-5}{5} \rceil} O(1) + \lceil \frac{n-5}{5} \rceil n - \frac{(\lceil \frac{n-5}{5} \rceil - 1)(\lceil \frac{n-5}{5} \rceil)}{2} 5$

$$= \Theta(3^{\frac{n}{5}})$$

4) $T(n) = 2T(\frac{n}{2}) + n^2 \qquad T(n) = 1$ when $n \le 1$.

$$T(n) = 2(2T(\tfrac{n}{4}) + (\tfrac{n}{2})^2) + n^2 = 2^k T(\tfrac{n}{2^k}) + (\tfrac{n}{2^{k-1}})^2 + \cdots + n^2.$$
$$= 2^k + (\tfrac{n}{2^{k-1}})^2 + (\tfrac{n}{2^{k-2}})^2 + \cdots + n^2.$$

where $\frac{n}{2^k} \le 1$ but $\frac{n}{2^{k-1}} > 1$

Then $\log_2 n \le k < \log_2 n + 1 \quad \therefore k = \log_2 n$

Then, $T(n) = 2^{\log_2 n} + (\frac{n}{2^{\log_2 n - 1}})^2 + \cdots + n^2.$

$\approx n + g_1(x) + g_2(x) + \cdots + g(x) + n^2.$ where c = constant

where $g_i(x) < n^2$ since

$2^{\log_2 n - i} > 1$ when $1 \le i < \log_2 n$

Then, $T(n) = \Theta(n^2)$

5) $T(n) = T\left(\frac{n}{5}\right) + T\left(\frac{7}{10}n\right) + n$ Let c_i be constant $\forall i$

$$T(n) \le T\left(\frac{n}{5}\right) + T\left(\frac{7}{10}n\right) + c_1 n$$

$$\le c_2\left(\frac{n}{5}\right)\log\frac{n}{5} + c_3\left(\frac{7n}{10}\right)\log\left(\frac{7}{10}n\right) + c_1 n$$

$$\vdots$$

$$= c_2' n\log n + c_3' n\log n + c_4 + c_1 n$$

$$= c_5 n\log n + c_1 n + c_4$$

$$\le c_6 n\log n.$$

$$\therefore T(n) = O(n\log n)$$

4

See hw-template.

5

1) In this case,

the algorithm simply calls the recursive call in line 5 with $n-1$ elements and in line 6 with 0 elements.

Also line 4 takes $\Theta(n-1)$ runtime, and other lines are constant.

Then, the runtime is $\Theta((n-1)(n-2)\cdots 1) = \Theta((n-1)!)$

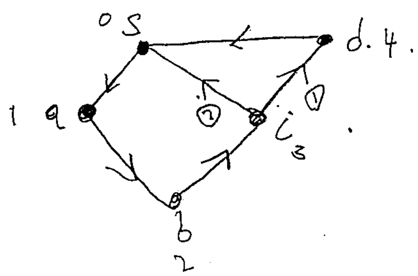
2) In this case

the algorithm simply calls the recursive call in line 5 with 0 element and in line 6 with $n-1$ elements.

Also line 4 takes $\Theta(n-1)$ runtime, and other lines are constant.

$\therefore$ the runtime is $\Theta((n-1)(n-2)\cdots 1) = \Theta((n-1)!)$

Consider the following graph G:



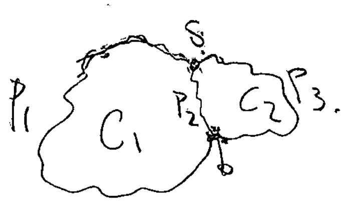
The arrows indicate the order of DFS exploration.

Starting at s,
we do following as we progress in the DFS explained:

level (s) = 0
level (a) = 1
level (b) = 2
level (c) = 3
level (d) = 4.
back edge (d s) ; cycle = 4 - 0 + 1 = 5
back edge (c s) ; cycle = 3 - 0 + 1 = 4

shortest cycle found = 4.
true shortest cycle = 3.

Now the problem of this algorithm occurs when two cycles in G overlaps in terms of vertices.



Let C_1, C_2 be cycles where $|C_2| < |C_1|$.
The problem is that the algorithm looks at the path P_1 and assume that P_1 is only path to b from s.
Then the algorithm ignores the cycle created by $P_2 \cup P_3$.