1. Indicate whether $f = O(g)$, or $f = \Omega(g)$, or both $f = \Theta(\,)$

a. $f = \Theta(g)$
b. $f = O(g)$
c. $f = \Theta(g)$
d. $f = \Theta(g)$
e. $f = \Theta(g)$
f. $f = \Theta(g)$
g. $f = \Omega(g)$
h. $f = \Omega(g)$
i. $f = \Omega(g)$
j. $f = \Omega(g)$
k. $f = \Omega(g)$
l. $f = O(g)$
m. $f = O(g)$
n. $f = \Theta(g)$
o. $f = \Omega(g)$
p. $f = O(g)$
q. $f = \Theta(g)$

2. Show that if c is a positive real number, then $g(n) = 1 + c + c^2 + \ldots + c^n$ is:

$$g(n) = 1 + c + c^2 + \ldots + c^n = \frac{c^{n+1} - 1}{c - 1}$$

a) If $c < 1$: $\lim_{n \to \infty} g(n) = \lim_{n \to \infty} \sum_{i=0}^{n} c^i = \frac{1}{1-c}$

If $c < 1$, the limit is constant. Therefore, $g = \Theta(1)$

b) If $c = 1$: $g(n) = 1 + c + c^2 + \ldots + c^n = \sum_{i=0}^{n} c^i = n+1 = \Theta(n)$

c) If $c > 1$: $\lim_{n \to \infty} \frac{g(n)}{c^n} = \lim_{n \to \infty} \frac{c^{n+1} - 1}{c^{n+1} - c^n}$

$$= \lim_{n \to \infty} \frac{c - \frac{1}{c^n}}{c - 1}$$

$\frac{1}{c^n}$ as n approaches $\infty = 0$

$$= \frac{c}{c-1}$$

$g(n) = \Theta(c^n)$ for $c > 1$

3. Is $4^{1536} - 9^{4824}$ divisible by 35?

Fermat's little theorem: For any prime p and $1 \leq a < p$, $a^{p-1} \equiv 1 \bmod p$.

$35 = 7 \cdot 5$, 5 and 7 are primes

$a^{5-1} \equiv 1 \pmod 5$ and $a^{7-1} \equiv 1 \pmod 7$

$(a^{5-1})^{7-1} = (a^4)^6 = a^{24} \equiv 1 \pmod{5 \cdot 7}$.

$a^{24} \equiv 1 \pmod{35}$ for all $1 \leq a < 35$

Therefore, $4^{1536} = 4^{24 \cdot 64} \equiv 1 \pmod{35}$ and $9^{4824} \equiv 9^{24 \cdot 201} \equiv 1 \pmod{35}$

$4^{1536} \equiv 9^{4824} \pmod{35}$. $\boxed{\text{Therefore, } 4^{1536} - 9^{4824} \text{ is divisible by } 35.}$

4. What is $2^{2^{2023}} \pmod 3$?

Fermat's little theorem: If p does not divide a, then $a^{p-1} \equiv 1 \pmod p$.

$2^{2n} \equiv 1 \pmod 3$

$2^{2n+1} \equiv 2 \pmod 3$

2^{2023} is even

$2^{\text{even}} \equiv 1 \pmod 3$

$\boxed{\text{Hence, } 2^{2^{2023}} \pmod 3 = 1}$

5. The most efficient way to calculate the nth Fibonacci number is to use matrices.

Write equations $F_1 = F_1$ and $F_2 = F_0 + F_1$ in matrix notation:

$$\begin{pmatrix} F_1 \\ F_2 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix} \cdot \begin{pmatrix} F_0 \\ F_1 \end{pmatrix}$$

Also write F_2 and F_3: $\begin{pmatrix} F_2 \\ F_3 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix} \cdot \begin{pmatrix} F_1 \\ F_2 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}^2 \cdot \begin{pmatrix} F_0 \\ F_1 \end{pmatrix}$

In general, $\begin{pmatrix} F_n \\ F_{n+1} \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}^n \cdot \begin{pmatrix} F_0 \\ F_1 \end{pmatrix}$

The number of operations needed is $O(\log n)$

The formula is $F_n = \frac{1}{\sqrt5}\left(\frac{1+\sqrt5}{2}\right)^n - \frac{1}{\sqrt5}\left(\frac{1-\sqrt5}{2}\right)^n$

And then you could calculate modulo 5 by finding the remainder of $F_n/5$.

6. $(\log n)^{\log}$ dominates $\frac{n}{\log n}$

Grad student B has the better algorithm as n goes to ∞.

7. The iterative algorithm takes $O((\log x)^2)$ for the first iteration. Then, it takes $O(i^2(\log(x))^2)$. So the time complexity is $O((\log x)^2 y^3)$. The recursive algorithm has a <u>better</u> time complexity: $O((\log_2(x))^2 y^2)$.

8. Find the inverse of: 20 mod 79; 3 mod 62; 21 mod 91; 5 mod 23

a. Inverse of 20 mod 79 is 4 mod 79

b. Inverse of 3 mod 62 is 21 mod 62

c. There is no inverse of 21 mod 91

d. Inverse of 5 mod 23 is 14 mod 23