

1. Compute

1. $\sum_{i=0}^n \sum_{k=0}^i 2^k$

$$\begin{aligned} \sum_{i=0}^n \sum_{k=0}^i 2^k &= \sum_{i=0}^n (2^{i+1} - 1) \\ &= \sum_{i=0}^n 2^{i+1} - \sum_{i=0}^n 1 \\ &= 2 \cdot \sum_{i=0}^n 2^i - \sum_{i=0}^n 1 \\ &= 2 \cdot (2^{n+1} - 1) - (n + 1) \end{aligned}$$

2. $4^{\log_2(5)} + \log^2(8^3)$

$$\begin{aligned} 4^{\log_2(5)} + \log^2(8^3) &= 2^{\log_2(5^2)} + 3^2 \cdot \log_2^2(8) \\ &= 25 + 3^2 \cdot 3^2 \\ &= 25 + 3^4 \end{aligned}$$

2. Use L'Hopital's rule to determine the limit of:

$$\begin{aligned} \lim_{x \rightarrow \infty} \frac{x \cdot \ln x^5 + 4}{(3 \cdot x + 1)^2} &= \lim_{x \rightarrow \infty} \frac{5 \cdot x \cdot \ln(x) + 4}{9 \cdot x^2 + 6 \cdot x + 1} \\ &= 0 \end{aligned}$$

3. What is the definition of $f(n) = \Omega(g(n))$.

$$f(n) = \Omega(g(n)) \iff \exists c > 0, \exists k \geq 0 \text{ s.t. } f(n) \geq c \cdot g(n) \forall n \geq k$$

4. What is the growth of the below function

$$f(n) = n \cdot 2^{\log_2(n)} + \sqrt{n} + 5 \cdot n \cdot \log^3(n) + \log^2(n^n)$$

1. $\Theta(n^2)$

2. $\Theta(n \cdot \log^3(n))$

3. $\Theta(\sqrt{n})$
4. $\Theta(n^2 \cdot \log^2(n))$
5. Neither!

$$\begin{aligned}
 f(n) &= n \cdot 2^{\log_2(n)} + \sqrt{n} + 5 \cdot n \cdot \log^3(n) + \log^2(n^n) \\
 &= n^2 + \sqrt{n} + 5 \cdot n \cdot \log^3(n) + n^2 \cdot \log^2(n) \\
 &= \Theta(n^2 \cdot \log^2(n))
 \end{aligned}$$

(4) is the correct answer.

5. What is the growth of the below function

$$f(n) = 6 \cdot \log(n) + 4 \cdot \log^4 \log^3(n) + 3 \cdot \log(n^2) + \log^2(n)$$

1. $\Theta(\log(n))$
2. $\Theta(\log(n^2))$
3. $\Theta(\log^4 \log^3(n))$
4. $\Theta(\log^2(n))$
5. Neither!

$$f(n) = \Theta(\log^2(n))$$

(4) is the correct answer.

8. Prove that $f(n) = 2 \cdot n \cdot \log(n^2) - 6 \cdot \log^2(n) + \sqrt{n}$ is $O(n \cdot \log(n))$, provide the appropriate C and k constants.

$$\begin{aligned}
 2 \cdot n \cdot \log(n^2) - 6 \cdot \log^2(n) + \sqrt{n} &= 4 \cdot n \cdot \log(n) - 6 \cdot \log^2(n) + \sqrt{n} \\
 &\leq 4 \cdot n \cdot \log(n) + 6 \cdot n \cdot \log(n) + \sqrt{n} \\
 &\leq 4 \cdot n \cdot \log(n) + 6 \cdot n \cdot \log(n) + n \cdot \log(n) \\
 &= 11 \cdot n \cdot \log(n), n \geq 4
 \end{aligned}$$

Let $c = 11$ and $k = 4$ then the statement translates to

$$2 \cdot \log(n^2) - 6 \cdot \log^2(n) + \sqrt{n} \leq c \cdot n \cdot \log(n), n \geq k$$

which by the definition of big- O notation, $2 \cdot n \cdot \log(n^2) - 6 \cdot \log^2(n) + \sqrt{n}$ is $O(n \cdot \log(n))$.

9. Compare the growth of $f(n) = 4^{2^{\log_2(n)}}$ and $g(n) = 2^{n+\log(n)}$

$$\begin{aligned} f(n) &= 4^n \\ g(n) &= n \cdot 2^n \end{aligned}$$

$$\begin{aligned} \lim_{n \rightarrow \infty} \frac{2^{2 \cdot n}}{n \cdot 2^n} &= \lim_{n \rightarrow \infty} \frac{2^n}{n} \\ &= \infty \end{aligned}$$

$f(n) = \omega(g(n))$ this implies $f(n) = \Omega(g(n))$. We also know then that $g(n) = O(f(n))$.

10. Prove transitivity of big- O : if $f(n) = O(g(n))$, and $g(n) = O(h(n))$, then $f(n) = O(h(n))$.

Since $f(n) = O(g(n))$ and $g(n) = O(h(n))$ we have the equalities

$$\begin{aligned} f(n) &\leq c_1 \cdot g(n), n \geq k_1 \\ g(n) &\leq c_2 \cdot h(n), n \geq k_2 \end{aligned}$$

From this we obtain

$$f(n) \leq c_1 \cdot c_2 \cdot h(n), n \geq k'$$

where $k' = \max(k_1, k_2)$. Let $c = c_1 \cdot c_2$ and $k = k'$ the statement then translate to

$$f(n) \leq c \cdot h(n), n \geq k$$

which by the definition of O notation, $f(n) = O(h(n))$.

11. What is the growth of $n^2 + 2 \cdot n^2 + 3 \cdot n^2 + 4 \cdot n^2 + \dots + n^5$?

$$\begin{aligned} \sum_{i=1}^{n^3} i \cdot n^2 &= n^2 \cdot \sum_{i=1}^{n^3} i \\ &= n^2 \cdot \frac{(n^3) \cdot (n^3 + 1)}{2} \\ &= \frac{n^8 + n^5}{2} \\ &= \Theta(n^8) \end{aligned}$$

12. Prove $\forall k > 0, \epsilon > 0 \implies \log^k(n) = o(n^\epsilon)$.

We prove this by induction on k , let ϵ be a real number greater than 0. When $k = 1$ we have

$$\begin{aligned} \lim_{n \rightarrow \infty} \frac{\log(n)}{n^\epsilon} &= \lim_{n \rightarrow \infty} \frac{\frac{1}{n \cdot \ln(2)}}{\epsilon \cdot n^{\epsilon-1}} \\ &= \lim_{n \rightarrow \infty} \frac{1}{\epsilon \cdot \ln(2) \cdot n^\epsilon} \\ &= 0 \end{aligned}$$

Assume that the result holds for all integers n less than or equal to some integer k , we show it must also hold for the $k + 1$ integer.

$$\begin{aligned} \lim_{n \rightarrow \infty} \frac{\log^{k+1}(n)}{n^\epsilon} &= \lim_{n \rightarrow \infty} \frac{(k+1) \cdot \log^k(n) \cdot \frac{1}{n \cdot \ln(2)}}{\epsilon \cdot n^{\epsilon-1}} \\ &= \lim_{n \rightarrow \infty} \frac{(k+1) \cdot \log^k(n)}{\epsilon \cdot \ln(2) \cdot n^\epsilon} \\ &= 0 \end{aligned}$$

which is what we wanted to show. Hence by the principle of mathematical induction the result holds for all $n > 0$ and $\epsilon > 0$.

13. Use the definition of big- O to prove that $f(n) + g(n) = O(\max(f(n), g(n)))$.

$$f(n) + g(n) \leq 2 \cdot \max(f(n), g(n)), n \geq 1$$

which by the definition of big- O notation, $f(n) + g(n) = O(\max(f(n), g(n)))$.

14. Prove or disprove $(n!)^{\log(n)} = \omega(n^{2^{\log(n)}})$

$$\begin{aligned} \log\left(\lim_{n \rightarrow \infty} \frac{n!^{\log(n)}}{n^n}\right) &= \lim_{n \rightarrow \infty} \log(n) \cdot \log(n!) - n \cdot \log(n) \\ &= \lim_{n \rightarrow \infty} \Theta(n \cdot \log_2^2(n)) - n \cdot \log_2(n) \\ &= \infty \\ 2^\infty &= \infty \end{aligned}$$

which by the definition of little- ω notation, $n!^{\log(n)} = \omega(n^{2^{\log(n)}})$.

15. Given a sorted array of n distinct number where the range of the numbers are between 0 and m and $m > n$. Find the smallest number.

```
1 #include <iostream>
2 #include <vector>
3
4 using namespace std;
5
6 // T(n) = T(n / 2) + O(1)
7 // T(n) = O(log2(n))
8 int binary_search(const vector<int>& A, int start, int end){
9     if(start > end) return end + 1;
10
11     if(A[start] != start) return start;
12
13     int mid = (start + end) / 2;
14
15     if(A[mid] == mid) return binary_search(A, mid + 1, end);
16
17     return binary_search(A, start, mid - 1);
18 }
19
20 int main() {
21     vector<int> A1 = {0, 1, 3, 6, 8, 9};
22     vector<int> A2 = {0, 2, 5, 7, 11};
23     cout << binary_search(A1, 0, A1.size() - 1) << endl;
24     cout << binary_search(A2, 0, A2.size() - 1) << endl;
25 }
```
