Problem 1

Yes, the mentioned problem can be solved in O(log n) time. The given array is A[1,...,n] having each item from (0,...,9). Moreover, A[i] < = A[i+1] for all i if we take an array with having values : [1,2,3,3,5,6,6,6,9] and if we use binary search to find the indexes of the repetitive elements, we will get the start and end index for each repetitive item in the array and using that, we can get the count for that number and sum them up for find the total sum of the elements of the array. for example, for 1: start index = end index = 0 => 0 - 0 + 1 => count = 1 for 2: start index = end index = 1 => 1 - 1 + 1 => count = 1 and so on and after that, we can calculate the sum as well. So, the total time complexity of this task => time complexity of binary search = O(log n)

Problem 2a

$$f(n) = n^{1.01}$$
$$g(n) = n(\log n)^2$$

$$f(n) = n^{1.01} = n^1 \times n^{0.01}$$

$$\lim_{n \to \infty} \frac{f(n)}{g(n)} = \frac{\cancel{n} \times n^{0.01}}{\cancel{n} \times (\log n)^2}$$

diffrentiate both the sides :

$$\frac{0.01 \times n^{-0.99}}{2 \log n \cdot \frac{1}{n}} = \frac{0.01 \times n^{0.01}}{2 \log n}$$

differentiate both the sides :

$$\frac{(0.01)^2 \; n^{-0.99}}{2 \cdot \frac{1}{n}} = \frac{(0.01)^2 \cdot n^{0.01}}{2}$$

$$= c \times n^{0.01} \quad \text{where } c = 0.0005$$

Hence, when $n \to \infty$ $\lim_{n \to \infty} \frac{f(n)}{g(n)} \to \infty$

Hence, $f(n) = \Omega(g(n))$

Problem 2b

$$f(n) = \frac{n^2}{\log(n)}$$

$$g(n) = n(\log n)^2$$

$$\lim_{n \to \infty} \frac{f(n)}{g(n)} = \frac{n^2}{\log(n) \cdot n(\log n)^2}$$

$$= \frac{n}{(\log n)^3}$$

differentiate both the sides.

$$= \frac{1}{3(\log n)^2 \cdot \frac{1}{n}} = \frac{n}{3(\log n)^2}$$

differentiate both the sides

$$= \frac{1}{3 \cdot 2 (\log n) \cdot \frac{1}{n}} = \frac{n}{6 \log n}$$

differentiate both the sides.

$$\frac{1}{6 \cdot \frac{1}{n}} = \frac{1}{6} n$$

Hence, $n \to \infty$ $\lim_{n \to \infty} \frac{f(n)}{g(n)} \to \infty$

hence, $f(n) = \Omega\, g(n)$

Problem 3a

$f(n) = (\log n)^{\log n}$

$g(n) = 2^{(\log n)^2}$

let $\log n = x$

$\therefore f(n) = x^x$

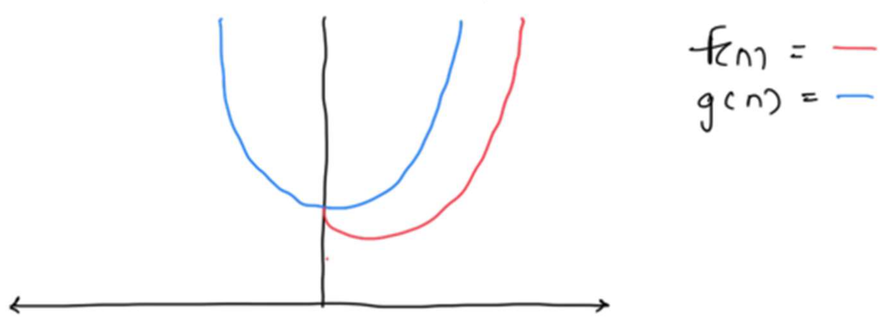
$g(n) = 2^{x^2} = (2^x)^x$

The exponents of both the functions are the same i.e x^x.

Hence, if we compare the base,

$f(n) = O\, g(n)$

If we plot these functions, we will get



Hence, from the graph $f(n) = O\, g(n)$

Problem 3b

$$f(n) = \sum_{i=1}^{n} i^k$$
$$g(n) = n^{(k+1)}$$

$$\sum_{i=1}^{n} i^k = 1^k + 2^k + \dots + n^k$$
$$n^{(k+1)} = n \cdot n^k = n^k + \dots \quad (n \text{ times})$$

$$1^k + 2^k + \dots + n^k \leq n \cdot n^k$$
$$\therefore \quad f(n) \leq g(n)$$

Hence, the upper bound is prooved.

For lower bound :

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

Here, half of the terms are $\geq \frac{n}{2}$

$$\therefore \quad f(n) \geq \left(\frac{n}{2}\right)^k + \left(\frac{n}{2}\right)^k + \dots \quad \frac{n}{2} \text{ times.}$$
$$\geq \frac{n}{2} \cdot \left(\frac{n}{2}\right)^k$$
$$\geq \left(\frac{n}{2}\right)^{k+1}$$
$$\geq \frac{n^{k+1}}{2^{k+1}}$$
$$\geq c \cdot n^{k+1} \quad \text{where } c = 2^{k+1}$$

Hence, $f(n) = \theta g(n)$