CPS130 : Homework 1 Solutions

1. Prove using induction that $\forall n > 0$,

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

**Proof** $P(n) : \sum_{i=1}^{n} i^3 = \left(\frac{n(n+1)}{2}\right)^2$

For $n = 1$,
LHS = $\sum i = 1^1 i^3 = 1 = \text{RHS}$
Thus $P(1)$ is true....(1)
Let us assume that $P(k)$ is true for some $k \geq 1$....(2)
By (2), we have
$$\sum_{i=1}^{k} i^3 = \left(\frac{k(k+1)}{2}\right)^2$$
Adding $(k + 1)^3$ to both sides,
$$\sum_{i=1}^{k+1} i^3 = \left(\frac{k(k+1)}{2}\right)^2 + (k + 1)^3$$
$$= (k + 1)^2\left(\frac{k^2}{4} + k + 1\right)$$
$$= (k + 1)^2\left(\frac{k^2 + 4k + 1}{4}\right)$$
$$= \left(\frac{(k+1)(k+2)}{2}\right)^2$$
which is the same as the RHS of $P(k+1)$
We have shown that $P(k) \Rightarrow P(k+1)$....(3)
Hence by (1), (2) and (3), using induction hypothesis, we can say that $P(n)$ is true $\forall n \geq 1$

2. * For each pair of functions $(f,g)$ given below, determine whether $f = O(g)$ and/or $f = \Omega(g)$ and/or $f = \Phi(g)$

(a) $2^n, e^n$
    $2^n = O(e^n)$ as $2 < e$

(b) $n^k, c^n$ where $k$ and $c$ are constants
    case 1: $c > 1$
    $n^k = O(c^n)$
    case 2: $c <= 1$
    $n^k = \Omega(c^n)$

(c) $n, 2^{\log_3 n}$
    $n = \Omega(2^{\log_3 n})$

(d) $(\log n)!$, $\log(n!)$
    $(\log n)! = \Omega(\log(n!))$ [Hint: Stirling’s Approximation]
3. 3.1-7 [CLRS pg 50]
Let us prove it by contradiction. Let \( f \in o(g(n)) \cap \omega(g(n)) \). That means, \( f = o(g(n)) = \omega(g(n)) \). By definition of \( o \), we have
\[
\lim_{n \to \infty} \frac{f(n)}{g(n)} = 0
\]
but by the definition of \( \omega \), we have
\[
\lim_{n \to \infty} \frac{f(n)}{g(n)} = \infty
\]
Since \( 0 \neq \infty \), we have an obvious contradiction.

4. Solve the following recurrences

(a) \( T(n) = T(n - 1) + n \)
\( T(n) = O(n^2) \)

(b) \( T(n) = 2T(n/4) + \sqrt{n} \)
\( T(n) = O(\sqrt{n \log n}) \)

(c) \( T(n) = T(9n/10) + n \)
\( T(n) = O(n) \)

(d) \( T(n) = 7T(n/2) + n^2 \)
\( T(n) = O(n^{\log_2 7}) \)

(e) \( T(n) = 3T(n/2) + n \log n \)
\( T(n) = O(n^{\log_2 3}) \)

5. * 4-2 [CLRS pg 85]
You first examine the least (or most, the argument will be symmetric) significant bit of each number. This will take \( O(n) \) steps. While examining the bits keep count of the number of 1’s and 0’s. Select those numbers which have a lesser count. The respective digit (1/0) will be the LSB of the missing number. Now look at the second least significant bit in the selected set. The size of the set can not be more than half the original set at every iteration.
The recurrence will be:
\[
T(n) = T(n/2) + n
\]
which can be solved to get \( T(n) = O(n) \)