DATE

HW: 1

CS 6212 - Design and Analysis of Algorithms.

Question 0:

(i) Σi, summation for $i = 1$ to n

Answer: $\Rightarrow 1 + 2 + \cdots + n \Rightarrow \frac{n(n+1)}{2}$

(ii) Σi^3, Summation for $i = 1$ to n

Answer: $\Rightarrow 1 + 8 + \cdots + n^3 \Rightarrow \left(\frac{n(n+1)}{2}\right)^2$

(iii) $\Sigma 2^i$, Summation for $i = 1$ to n

Answer $\Rightarrow 2^{n+1} - 2$

(iv) $\Sigma 1/i$, Summation for $i = 1$ to n

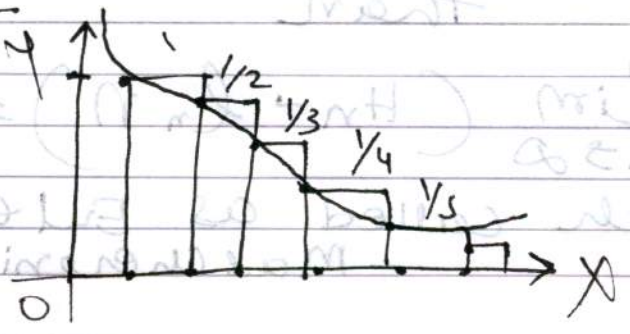
→ Answer: $\Rightarrow 1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{n}$

it is harmonic series.

We can estimate as below:

if we take

→ $H_n - 1/n = 1 + 1/2 + 1/3 + \cdots + 1/n+1$

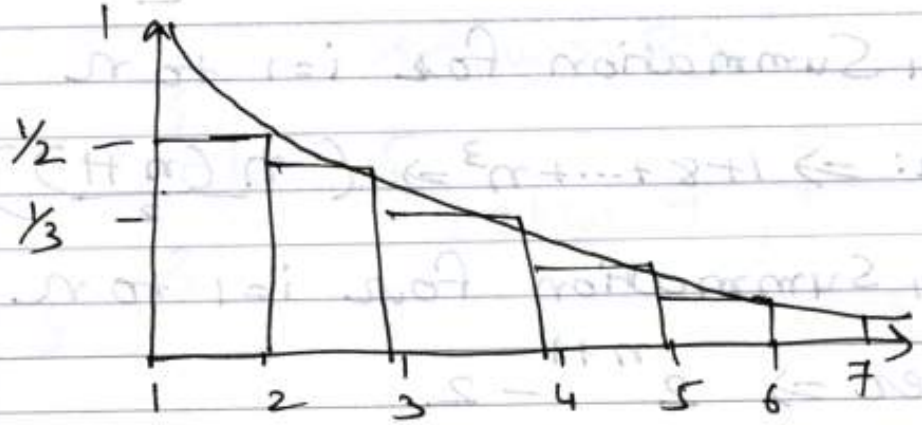


And for

$H_{n-1} = 1 + 1/2 + 1/3 + \cdots + 1/n$

$H_{n-1} = \frac{1}{2} + 1/3 + \cdots + 1/n$

which overestimated of the area



Hence

$$H_{n-1} < \int_1^n \frac{1}{n}\,dx < H_n - 1/n$$

$$\ln n + 1/n < H_n < \ln n + 1$$

and if we take Euler Property than

$$\lim_{n \to \infty} (H_n - \ln n) = \gamma$$

γ which called as Euler - Mascheroni Constant

Question 1 : PMI

① $1^2 + 2^2 + 3^2 + \cdots + n^2 = \frac{n(n+1)(2n+1)}{6}$

Answer To prove the statement, we need to show following cases.

(i) Base Case: (Show $n=1$ is true)
(ii) Inductive Hypothesis (Assume $n=k$ is true)
(iii) Inductive step (show that $n=k$ is true then $n=k+1$ is also true.)

(i)
For Base Case 1: $1^2 + 2^2 + 3^2 + \cdots + n^2 = \frac{n(n+1)(2n+1)}{6}$

$\Rightarrow$ take $n=1$

$$(1)^2 \stackrel{?}{=} \frac{1(1+1)(2(1)+1)}{6}$$

$$\stackrel{?}{=} \frac{1(2)(3)}{6} = 1$$

Base case is checkout. and its true.

(ii)
For Inductive Hypothesis (Assume $n=k$ is true)

Assume that $1^2 + 2^2 + 3^2 + \cdots + k^2 = \frac{k(k+1)(2k+1)}{6}$

is true. for sum k in the set of positive integers.

(iii) Inductive step: (Show that $n=k$ is true $\Rightarrow n=k+1$ is also true)

Let's take $\underbrace{1^2+2^2+\dots+k^2}+(k+1)^2$

$$= \frac{k(k+1)(2k+1)}{6} + (k+1)^2$$

(By Assume)

$$= \frac{k+1}{6}\left[k(2k+1)+(k+1)\right]$$

$$= \frac{k+1}{6}\left[2k^2+2k+6k+6\right]$$

$$= \frac{k+1}{6}\left[2k^2+8k+6\right]$$

$$= \frac{k+1}{6}(k+2)(2k+3)$$

So, the statement is true for $n=1$ and also true for $n=k$ so that $n=k+1$ is also true for the statement is true for all positive N.

(2) $1^3 + 2^3 + 3^3 + \cdots + n^3 = \left[\frac{n(n+1)}{2}\right]^2$

Answer: To prove the statement, we need to prove following cases.

(i) Base Case (Show $n=1$ is true)

$$1^3 + 2^3 + \cdots + n^3 = \left[\frac{n(n+1)}{2}\right]^2$$

take $n=1$

$$(1)^3 \stackrel{?}{=} \left[\frac{1(1+1)}{2}\right]^2 = (1)^2 = 1$$

$\Rightarrow$ Basecase is true.

(ii) Inductive Hypothesis (Assume that $n=k$ is true)

$$\Rightarrow 1^3 + 2^3 + 3^3 + \cdots + k^3 = \left[\frac{k(k+1)}{2}\right]^2$$

$$= \frac{1}{4} k^2 (k+1)^2$$

is true for sum of k in the set of positive Integer.

(iii) Inductive step: (Show that $n=k$ is true $\Rightarrow$ $n=k+1$ is also true.)

$$1^3 + 2^3 + 3^3 + \cdots + k^3 + (k+1)^3$$

$$= \frac{1}{4}k^2(k+1)^2 + (k+1)^3$$

$$= \frac{(k+1)^2}{4}\left[k^2 + 4(k+1)\right]$$

$$= \frac{(k+1)^2}{4}\left[k^2 + 4k + 4\right]$$

$$= \frac{1}{4}(k+1)^2\left[(k+2)^2\right]$$

$$= \frac{1}{4}(k+1)^2\left[(k+1)+1\right]^2$$

* So the statement is true for $n=1$ and truth for $n=k$ implies that $n=k+1$ is also true so, the statement is true for all positive Integers.

$\Rightarrow$ So that it is proved through PMI.

DATE

Question 2 - Asymptotic Notation

① $2^n = \omega(n^k)$

⇒ ~~its~~ inver ω is inverse of o
(small ~~om~~ oh notation)

That is, $f(n) = o(g(n))$ if and only if $g(n) = \omega(f(g)(n))$

Thus,

⇒ $n^k = o(2^n)$

$\exists n_0, \forall c > 0 \quad \lim_{n \to \infty} \frac{f(n)}{g(n)} = 0$

Let's take L'Hopital's Rule.

$\lim_{n \to \infty} \frac{n^k}{2^n}$

~~tak ln e both~~

$\lim_{n \to \infty} \frac{k \ln n^n}{n \ln n^2}$ take ln.

where k = Constant

$\lim_{n \to \infty} \frac{k}{n \ln n_e^2}$ $k\left(\frac{1}{\infty}\right)$ $k \cdot 0$

$= 0$

So that, we can conclude that

$$n^k = O(2^n)$$

~~in same way~~

Such that

$$\Rightarrow \quad 2^n = \omega(n^k) \quad \text{// inverse of o notation}$$

(2) $(1+2+3+\ldots+n) + (1^2+2^2+\ldots+n^2)$

$$= O(n^3)$$

$$\frac{n(n+1)}{2} + \frac{n(n+1)(2n+1)}{6} = O\,n^3$$

$\exists n_0, c$ such that

$$f(n) \leq c\,g(n) \quad \forall n \geq n_0$$

$$\frac{n(n+1)}{2} + \frac{n(n+1)(2n+1)}{6} \leq c\,n^3$$

$$\frac{n(n+1)}{6}\left[3 + 2n+1\right] \leq c\,n^3$$

$$\frac{n(n+1)}{6}\left[2n+4\right] \leq c\,n^3$$

$$\frac{n(n+1)(n+2)}{3} \leq cn^3$$

$$\frac{(n^2+n)(n+2)}{3} \leq cn^3$$

$c = 20$

$n_0 = 2$

So, $\exists n_0 = 2$ $\exists c = 100$, we can prove that

$$(1+2+\dots+n)+(1^2+2^2+\dots+n^2) = O(n^3)$$

(3) $\log(n!) = \Theta(n \log(n))$

Take ~~if~~ $f(n) = \log(n!)$

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

$f(n) = \Theta(g(n))$

if and only $f(n) = O(g(n))$

and $g(n) = O(f(n))$

Let's first prove

$f(n) = Og(n)$ Such that

$\log n! \leq c(n \log n)$

$\log 1 + \log 2 + \dots + \log n \leq c(n \log n)$

Here $\exists n_0 > 2$

$\exists c = 100000$

Such that $n \geq n_0$

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

And

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

$n \log n \leq c \log n!$

Here $\exists n_0 > 2$

$\Rightarrow c = 100000$

Such that $\forall n \geq n_0$

$g(n) = \Theta f(n)$

Such that $f(n) = \Theta(g(n))$

(4) $\log(1^2) + \log(2^2) + \ldots + \log(n^2)$

$= \Theta(n \log(n))$

Here $\exists n_0, c$ such that

$f(n) \leq c\, g(n)$

$\log 1^2 + \log 2^2 + \ldots + \log n^2 \leq c\, n \log n$

we can take for $\exists c$ and $\exists n_0$ where

$n \geq n_0$

$c = 1000$

$n_0 \geq 5$

We can prove

$f(n) \leq g(n)$

$\log(1^2) + \log(2^2) + \ldots + \log n^2 = O(n \log(n))$

Question 3 - A Tennis Tournament

Answer : It is impossible to find the largest number given n number using less than N-1 Comparision.

⇒ Let's take an Example :

if we have 5 number in array

5	4	3	2	8

~~and~~ so $n = 5$

For n-1 less Comparision we have to take as per Algorith

$j = 1$ to $n-2$

So, if we inlist it then

$x = a[0] = 5$

① | 5 | 4 | 3 | 2 | 8 | for $j = 1$

$a[0] \rightarrow x$

$a[i]$

$a[j]$

⇒ So, ~~x =~~ x is not less than $a[j]$ So that the for loop will increment to 2

So, here we are taking N-2 Comparision and the last term is greater than every element. Thus, we can not compare betn

N and N-4 term. So Here we can't find actual Highest number among them.

- So, when the size of array is n then to find largest number we have to take n-1 Comparision. ~~Not~~ So, For n-1 comparision, it would not work.

Question 4: Coins:

Algorithm: (Weighing 3 times)

Case I: Weigh(1,2,3,4) and (5,6,7,8);

```
if (wt(1,2,3,4) == wt(5,6,7,8)) {
    weigh((9,10) & (1,2));
    if (wt(9,10) == wt(1,2)) {
        weigh (11 and 1);
        if (wt(11) == wt(1)) {
            return 12;
        } else { return 11; }
```

```
else {
    weigh (9 and 1);
    if (wt(9) == wt(1)) {
            return 10; }
        else {
            return 9;
        }
    }
else if (wt(1,2,3,4) > wt(5,6,7,8))
    weigh((5,1,2) and (6,3,4));
    if (wt(5,1,2) == wt(6,3,4)){
        weigh (7 and 9);
        if (wt(7) == wt(9)) {
            return 8;
        } else
            { return 7;
        }
} else if (wt(5,1,2) > wt(6,3,4))
    weigh (1 and 2)
    if (wt(1) > wt(2)) {
        return 1;
    }
    else if (wt(2) > wt(1)) {
        return 2;
    } else {
```

```
return 6;
}
else {
 weigh (3 and 4)
  if ( wt(3) > wt(4)) {

        return 3; }
else  if ( wt(3) < wt(4)) {

        return 4; }
    else
        {
          return 5;
     }
}

else {
weigh ((1,5,6) and (2,7,8));
   if ( wt(1,5,6) == wt (2,7,8)){
      weigh (5 and 6).
     if ( wt(5) > wt(6)) {

          return 5;
      } else if ( wt(6) > wt(5)){

          return 6; }
     else {
          return 2;
      }
```

```
else {
    weigh(7 and 8)
    if ( wt(7) > wt(8) ) {
        return 7;
    }
    else if (wt(7) < wt(8)) {
        return 8;
    }
    else {
        return 1;
    }
}
}
```

~~I think~~

* We can't find algorithm that determine the different coin in at most 2 weighings.

———x———