HW Assignment 1:
Ex: 3, 5, 6, 7, 8
DUE DATE: ~~29 June~~ 30 June

HW Assignment 2:
Ex 9, 10, 11, 12
Due date: ~~1 July~~ 2 July

Jun 22-1:12 PM

- Midterm 1: 6 July
- Midterm 2: 20 July

Jun 22-1:24 PM

$n! = n \times (n-1)!$

$(n-1)(n-2)!$

$(n-2)(n-3)!$

$\vdots$

1

Jun 22-1:46 PM

{ if (n = 0) return 1;
else return $F(n-1) \times n$;
}

$M(n) = 1 + M(n-1)$

return $5 \times F(4)$
$4 \times F(3)$
$3 \times F(2)$
$2 \times F(1)$
$1 \times F(0)$

return: 120

Jun 22-1:47 PM

$M(n) = 1 + M(n-1)$

$M(n) = 1 + (1 + M(n-2))$

$M(0) = 0$

$= 2 + M(n-2)$

$M(n) = 2 + (1 + M(n-3))$

$= 3 + M(n-3)$

$M(n) = i + M(n-i)$

$i = n$

$M(n) = n + M(0) = n$.

Jun 22-1:53 PM

$\begin{cases} M(n) = 2M(n-1) + 1 \\ M(1) = 1 \end{cases}$

$M(n) = 2[2M(n-2) + 1] + 1$

$= 2^2 M(n-2) + 3 = (2^2 - 1)$

$= 2^2[2M(n-3) + 1] + 3$

$= 2^3 M(n-3) + 7$

$= 2^3 M(n-3) + 2^3 - 1$

$= 2^i M(n-i) + 2^i - 1$

$n - i = 1,\ i = n-1$

$= 2^{n-1} M(1) + 2^{n-1} - 1$

$= 2^{n-1} + 2^{n-1} - 1$

$= 2 \times 2^{n-1} - 1 = 2^n - 1$

Algorithm Time complexity $O(2^n)$

Jun 22-2:00 PM

Bin(n)
{ if n = 1 return 1
  if n = 0 return 0
  else return $\left[\text{Bin}\left(\frac{n}{2}\right)\right] + 1$

$3 = 11 \quad (2^2 - 1)$

$5 \,|\, 2 \rightarrow \boxed{1}$, $2 \,|\, 2 \rightarrow \boxed{0}$, $1 \,|\, 2 \rightarrow \boxed{1}$, 0

$7 = 111$

$8 = 1000$

$n = 1024$

$\rightarrow 1023 = 1 \ldots 1$

0	0	0
n	n	n

Jun 22-2:09 PM

$3 = 2^2 - 1 \quad 8 = 2^3$

$\textcircled{7} = 2^{\textcircled{3}} - 1 \quad 3 = \log_2 8$

$15 = 2^4 - 1$

$$\begin{cases} \Pi(n) = \Pi\left(\frac{n}{2}\right) + 1 \\ \Pi(1) = 0 \end{cases}$$

$n = 2^k \quad k \geq 0$

$\Pi(2^k) = \Pi(2^{k-1}) + 1$

$= (\Pi(2^{k-2}) + 1) + 1$

$= \Pi(2^{k-2}) + 2$

$= (\Pi(2^{k-3}) + 1) + 2$

$= \Pi(2^{k-3}) + 3$

$= \Pi(2^{k-i}) + i$

$\Pi(0) \quad 2^{k-i} \Rightarrow \Pi(1), \Pi(0)$

$\Pi(1) \quad 2^{k-i} = 1 \Rightarrow k = i$

$= \Pi(1) + k = k$

$n = 2^k, \; k = \log_2 n$

$\Theta(\log_2 n)$

Jun 22-2:17 PM

function S(n)
{ if n = 1 return 1;
  else return $S(n-1) + n \times n \times n$;
}

- what does this algorithm do?
- Calculate $\Pi(n)$?

$n = 5$

$S(5) = S(4) + 5^3$

$= (S(3) + 4^3) + 5^3$

$= (S(2) + 3^3) + 4^3 + 5^3$

$= S(1) + 2^3 + 3^3 + 4^3 + 5^3$

$= 1^3 + 2^3 + 3^3 + 4^3 + 5^3$

$S(n) = \sum_{i=1}^{n} i^3$

Jun 22-2:35 PM

$$\begin{cases} \Pi(n) = \Pi(n-1) + 3 \\ \Pi(1) = 0 \end{cases}$$

$\Pi(n) = \Pi(n-1) + 3$

$= (\Pi(n-2) + 3) + 3$

$= \Pi(n-2) + 2 \times 3$

$= (\Pi(n-3) + 3) + 2 \times 3$

$= \Pi(n-3) + 3 \times 3$

$= (\Pi(n-4) + 3) + 3 \times 3$

$= \Pi(n-4) + 4 \times 3$

$\Pi(n) = \Pi(n-i) + i \times 3$

$n - i = 1$

$i = n - 1$

$\Pi(n) = \Pi(1) + (n-1) \times 3$

$= 0 + (n-1)3$

$\Pi(n) \in \Theta(n)$

Jun 22-2:49 PM

F(n)
{ if n = 1 return 1;
  else return F(n-1) + n
}

1) what does this algo do?
2) Calculate Time complexity of the algorithm?

$F(5) = F(4) + 5$

$= F(3) + 4 + 5$

$= F(2) + 3 + 4 + 5$

$= F(1) + 2 + 3 + 4 + 5$

$= 1 + 2 + 3 + 4 + 5$

For any n

$F(n) = \sum_{i=1}^{n} i$

$\Pi(n) = \Pi(n-1) + 1$

$M(1) = 0$

$M(n) = [M(n-2) + 1] + 1$

$= \Pi(n-2) + 2$

$= [\Pi(n-3) + 1] + 2$

$= \Pi(n-3) + 3$

$= \Pi(n-i) + i$

$n - i = 1 \Rightarrow i = n - 1$

$\Pi(1) + (n-1)$

$= 0 + (n-1) = n - 1$

$\Pi(n) \in \Theta(n)$

Jun 22-3:03 PM