Name: Sabbir Ahmed
Section: 02
HW #: 2
Version: A
Username: sabbir1

1. Prove $2^n > n^2$ for every positive integer n when n >= 5.

Let P(n) represent the proposition $2^n > n^2$, $n \in \mathbb{N}$ | n >= 5.

Let n = 5, so:

→ $2^5 > 5^2$

→ 32 > 25

→ P(n) to be true for n = 5.

Assume P(k) holds for all $k \in \mathbb{N}$ | k >= 5. So $2^k > k^2$.

Consider k+1. So:

→ $2^{k+1} > (k+1)^2$

≡ $2*2^k > k^2+2k+1$

≡ $2^k+2^k > k^2+2k+1$

But $2^k > k^2$

∴ $2^{k+1} > (k+1)^2$.

$2^n > n^2$ holds for all $n \in \mathbb{N}$ | n >= 5.

2. Prove $4^n - 1$ is divisible by 3 for all $n \in \mathbb{N}$.

Let P(n) represent the proposition $(4^n - 1) = 3k$ for all $n \in \mathbb{N}$.

Let n = 1. So:

→4^1-1

→3, which is divisible by 3.

→P(n) holds for n =1.

Assume P(a) holds for all $a \in \mathbb{N}$. So $(4^a - 1) = 3k$.

Consider a+1. So:

→$(4^{a+1} - 1)$

→$(4*4^a - 1)$

But $(4^a - 1) = 3k$

So, $(4*4^a – 1) \equiv (4*(3k + 1) - 1)$

$\rightarrow (4*3k + 4 - 1) \equiv (12k + 3) \equiv 3*(4k+1)$

$\therefore (4^{a+1} – 1)$ is divisible by 3.

$(4^n – 1) = 3k$ holds for all $n \in \mathbb{N}$.

3. Prove $\frac{1}{1*2} + \ldots + \frac{1}{n*(n+1)} = \frac{n}{n+1}$

Let P(n) represent the proposition $\frac{1}{1*2} + \ldots + \frac{1}{n*(n+1)} = \frac{n}{n+1}$

Let n = 1. So:

$\rightarrow \frac{1}{1*(1+1)} = \frac{1}{1+1}$

$\rightarrow \frac{1}{2} = \frac{1}{2}$

$\rightarrow$P(n) holds for n = 1.

Assume P(k) holds for all $k \in \mathbb{N}$. So $\frac{1}{1*2} + \ldots + \frac{1}{k*(k+1)} = \frac{k}{k+1}$

Consider k+1. Then:

$$\rightarrow \frac{1}{1*2} + \ldots + \frac{1}{k*(k+1)} + \frac{1}{(k+1)*(k+2)} = \frac{k}{k+1} + \frac{1}{(k+1)*(k+2)}$$

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

$$\rightarrow \frac{k+1}{k+2} \equiv \frac{(k+1)}{(k+1)+1}$$

$$\therefore \frac{1}{1*2} + \ldots + \frac{1}{k*(k+1)} + \frac{1}{(k+1)*(k+2)} = \frac{(k+1)}{(k+1)+1}$$

$\frac{1}{1*2} + \ldots + \frac{1}{n*(n+1)} = \frac{n}{n+1}$ holds for all $n \in \mathbb{N}$.

4.

Code	Cost	Times	Total
int sum = 0;	1	1	1
for (int i = 0; i < n; i++)	1 (int i) + n+1 (<) + n (++)	1	2n+2
for (int j = 0; j < n; j++)	1 (int j) + n+1 (<) + n (++)	n	$2n^2+2n$
++sum;	2	n*n	$2n^2$
FINAL	**$4n^2+4n+3 = O(n^2)$**		

5.

Code	Cost	Times	Total
int sum = 0;	1	1	1
for (int i = 0; i < n; i += 2)	1 (int i) + n+1 (<) + n/2 (+= 2)	1	n+n/2+2
for (int j = 0; j < n; j++)	1 (int j) + n+1 (<) + n (++)	n/2	n^2+n
++sum;	2	n*n/2	n^2
FINAL	**$2n^2+2n+n/2+3 = O(n^2)$**		

6.

Code	Cost	Times	Total
int sum = 0;	1	1	1
for (int i = 1; i < n; i *= 2)	1 (int i) + n+1 (<) + $\log_2(n)$ (*= 2)	1	$n+\log_2(n)+2$
for (int j = 0; j < n; j++)	1 (int j) + n+1 (<) + n (++)	$\log_2(n)$	$2n\log_2(n)+2\log_2(n)$
++sum;	2	$n*\log_2(n)$	$2n\log_2(n)$
FINAL	**$4n\log_2(n)+3\log_2(n)+n+3 = O(n\log_2(n))$**		

7.

Code	Cost	Times	Total
int sum = 0;	1	1	1
for (int i = 0; i < n; i++)	1 (int i) + n+1 (<) + n (++)	1	2n+2
for (int j = 0; j < i * i; j++)	1 (int j) + n^2+1 (< i*i) + n (++)	n	n^3+n^2+2n
for (int k = 0; k < j; k++)	1 (int k) + n^2+1 (< j) + n (++)	n^3	n^5+n^3+2n
++sum;	2	n^5	$2n^5$
FINAL	**$3n^5+2n^3+n^2+6n+3 = O(n^5)$**		