Problem 3 Shashank Agrawal

A8 - CS5008

1) For all x, there exists a 'y', such that $x+y$ equals 0

Yes 'x' & 'y' can be any integers.

If $x+y=0$, $-x=+y$. ~~(x is the ...~~

$\therefore$ $x-x$ $=0$; every x can have a negative value.(y).

2) There exists a y for every x, such that the sum of x & y is equal to x.

x & y can't be any integers in this case as y can only be 0.

$x+y=x$

$\Rightarrow y=0.$

The statement is still true as every integer x can be added to y, which is 0, to give itself (x).

3) There exists a 'x' for every 'y', such that the sum of x & y is equal to x

$x+y=x$

$y=0.$

This statement is false; as y can only take up the value of 0.

Problem 2

100	10000	3n	100n	$n\log_2 n$	$n\log_3 n$

100	3n	$n\log_2 n$	n^2	2^n	$n!$
10000	100n	$n\log_3 n$	$5n^2+3$		

Prob. 3

- Constant time $\rightarrow O(1)$
- Log. time $\rightarrow O(\log_2 n)$
- Linear time $\rightarrow O(n)$
- Quad. time $\rightarrow O(n^2)$
- Cubic time $\rightarrow O(n^3)$
- Exp. time $\rightarrow O(2^n)$
- Fac. time $\rightarrow O(n!)$

Prob. 4

1) $100n+5=O(2n)$

$$\frac{f(n)}{g(n)} = \frac{100n+5}{2n} < \frac{100n+5n}{2n} = \frac{105}{2} = 52.5$$

$C = 52.5$

Note : $5 < 5n$.

$\therefore$ $100n+5$ is $O(2n)$ as
$100n+5 \leq 52.5(2n)$ when $n>1$.

2) $n^3+n^2+n+100 = O(n^3)$

$$\frac{f(n)}{g(n)} = \frac{n^3+n^2+n+100}{n^3} < \frac{n^3+n^3+n^3+100n^3}{n^3} = 103$$

$C = 103$

$n^2 < n^3$, $n < n^3$, $100 < 100n^3$ for all $n>1$

$\therefore$ $n^3+n^2+n+100$ is $O(n^3)$ as
$n^3+n^2+n+100 \leq 103\, n^3$ for all $n>1$.

3) $n^{99} + 10000000 = O(n^{99})$

$$\frac{n^{99} + 10000000}{n^{99}} < \frac{n^{99} + 10000000n^{99}}{n^{99}} = 10000001$$

$C = 10000001$

$10000000 < 10000000n^{99}$ for all $n > 1$

$\therefore n^{99} + 10^7$ is $O(n^{99})$ as

$n^{99} + 10^7 \leq 1000001 \cdot n^{99}$ for all $n > 1$.

Prob. 4

1) 2048 steps

2) Binary search algorithm.

~~You~~ The algorithm compares the ~~middle~~ value in the middle index of the array & checks if the element to find is greater or lesser than the element in the middle index.

If lesser it ~~discards~~ checks in the array with end size (0, mid-1)

else checks in array size (mid+1, array.length -1)

else middle element is value

else value not found.

The binary search algo. cuts the array in half everytime it compares the middle element to the element to be found. ~~Big O(...)~~ & thus doesn't compare all array elements & is faster.

Big O: $O(\log n)$

3) 8 steps : $2^8 = 256$.

Array Size

256 → 128 Element not found

128 → 64 "

64 → 32 "

32 → 16 "

16 → 8 "

8 → 4 "

4 → 2 "

2 → 1 "

1

Prob. 5

a) You could use a binary search for this.

You take 50 coins & weigh them against ~~500z~~ 50oz.
If scale tips, your coin is in there. Cut number of coins by half $(50/2) = 25$ & repeat but weigh against 25oz. ~~Repeat~~

If your coin wasn't in the first 50 coins, take the other 50 coins & weigh those against 50oz.

Repeat & cut no. of coins & weight by half each time.
If odd no. like 25, split into 12 + 13.

b) ~~100 →~~ 50 → 25 → 12 → 6 → 3 → 2 → 1

3 → 1

$\boxed{\text{Max. 6 weighings}}$

25 → 13 → 6 → 3 → 2 → 1

3 → 1

13 → 7 → 4 → 2 → 1

7 → 3 → 2 → 1

3 → 1