Ques. 2

(a)

```
def find_complexity(n):
    i=1, s=1
    while s<=n:
        i+=1
        s+=i
    return
```

Ans. $\sqrt{n}$

Since, s is changing ~~exponentially~~ arithmetically the loop will only run half the time or less.

(b)

```
def find_complexity2(n):
    for i in range(n//2, n+1):
        for j in range(1, n-(n//2)+1):
            k=1
            while k<=n:
                k*=2
    return
```

for i loop → $n/2$

for j loop → $n/2$

while loop → $\log n$

Sol, $O(n^2 \log n)$

(c)

```
def find_complexity3(n):
    if(n<=2):
        return 1
    else:
        return find_complexity3(math.floor(math.sqrt(n))+1)
```

Ans. $\log(n)$

if $n=100$ ~~...~~ the program runs 4 times
& if $n=16$ the program runs 2 times
which are approximately
equal to $\log(n)$

(d)

```
def find-complexity4(n, epsilon):
    s, e = 1, n
    m = (s+e)/2

    while abs(n-m*m) > epsilon:
        if m*m > n:
            e = m
        else:
            s = m
        m = (s+e)/2
    return m
```

Ans. $O(n\log(n))$

Since the condition in ~~...~~ while loop is decreasing exponentially it will take $\log n$ time to execute and each iteration will execute operations n times.

Que. 5

- $f(n) = 5n^3 + 2n^2 + 7n + 1$

To prove $O(n^3)$

$$c \cdot g(n) = 5n^3 + 2n^3 + 7n^3 + n^3$$
$$= \underbrace{15}_{c}\,\underbrace{n^3}_{g(n)}$$

{if we use the highest power variable for all const. in the given f(n) then it gives the $c * g(n)$ which will always be $\geq f(n)$ }

now, to find n_0

$$5n^3 + 2n^2 + 7n + 1 \leq 15n^3$$

$$\frac{5n^3}{n^3} + \frac{2n^2}{n^3} + \frac{7n}{n^3} + \frac{1}{n^3} \leq 15$$

$$5 + \frac{2}{n} + \frac{7}{n^2} + \frac{1}{n^3} \leq 15$$

Let $n = 1$.

$$5 + \frac{2}{1} + \frac{7}{1^2} + \frac{1}{1^3} \leq 15$$

$$5 + 2 + 7 + 1 \leq 15$$

$$15 \leq 15$$

So, $5n^3 + 2n^2 + 7n + 1 \leq 15n^3$

$\forall \; n \geq 1$

$\therefore$ The upper bound is n^3

- $f(n) = 5n^3 + 2n^2 + 7n + 1$

To prove : $f(n) \in \Theta(n^3)$

$\because f(n) \in O(n^3)$
$f(n) \in \Omega(n^2)$
for $c = 15$ & $n > 1$

$\therefore \Theta(n) = O(n) \cap \Omega(n)$
$= n^3 \cap n^2$

$\Theta = n^3$

$f(n) \in \Theta(n^3)$

Ques. 5

- $f(n) = 5n^3 + 2n^2 + 7n + 1$

To prove $\Omega(n^2)$

$$g(n) = 5n^2 + 2n^2 + 7n^2 + n^2$$
$$= \underbrace{15}_{c} n^2$$

$$5n^3 + 2n^2 + 7n + 1 \geq 15n^2$$
$$5n + 2 + \frac{7}{n} + \frac{1}{n^2} \geq 15$$

$n = 1$

$$5 + 2 + 7 + 1 \geq 15$$
$$15 \geq 15$$

$$\therefore f(n) \in \Omega(n^2) \ \forall \ n_0 > 1$$

• $f(n) = 5n^3 + 2n^2 + 7n + 1$

To prove: $\omega(n^2)$

$$\lim_{n\to\infty}\left(\frac{n^2}{5n^3+2n^2+7n+1}\right) = 1$$

$$\lim_{n\to\infty}\left(\frac{1}{5n + 2 \cancel{+} 7/n + 1/n^2}\right) = 1$$

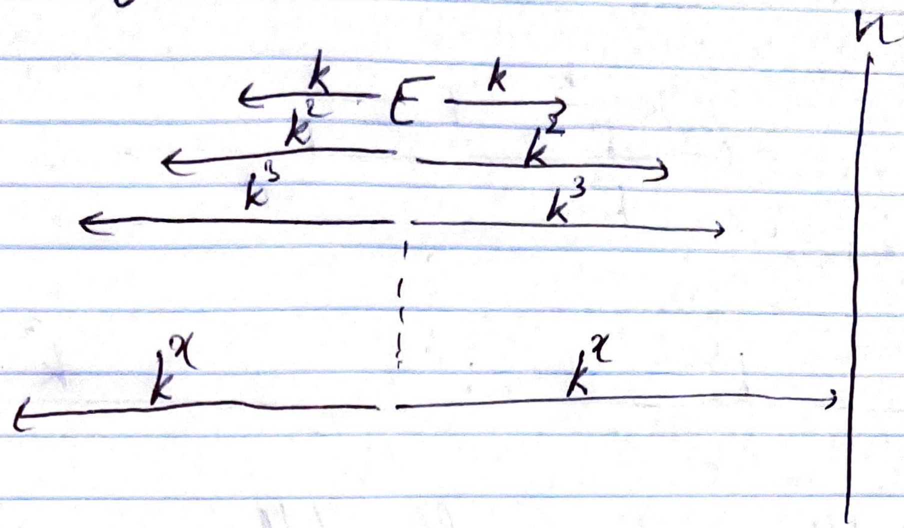
~~If $n =$ 1~~

$$\cancel{\lim_{n\to\infty}\left(\frac{1}{5+2+7+1}\right) = 1}$$

If $n \to \infty$ $5n$ will get really large & $7/n$, $1/n$ will get ~~sall~~ really small.

So, $\dfrac{1}{5n+2+7/n+1/n^2}$ will be approx. equal to 1

Ques ·6



He will walk till,

$$k^x = n$$

$$2k + 2k^2 + 2k^3 + \cdots + 2k^x$$

$$2k\left(\frac{k^x - 1}{k - 1}\right)$$

$$\frac{2k}{\underbrace{k-1}_{\text{constant}}}\left(n \underbrace{-1}_{\text{constant}}\right)$$

$\therefore$ Time complexity $= O(n)$