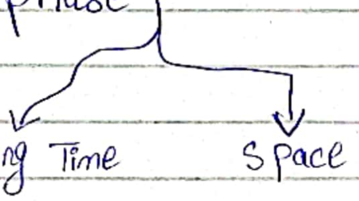
Design & Analysis of Algorithm

In case of any problem:

Steps:

1) Identify & write its algo
2) Translate it into Mathematical language (pseudo code)
3) Analysis phase



In running time, terms introduced are:

- Worst-case (Maximum running time of an algo on input size)

e.g. let code:

```
func(n)
{ for(i=n; i>1; i--)
      if (n==100)
         break; }
```

Let l be input size that varies so,

$$T_{worst}(l) = \underset{|l|=n}{Max}\left[T(l) \right]$$

- Average-case: Based on Probability

$$T_{avg}(I) = \sum_{i=1}^{n} P(I_i) \times T_i$$

Mathematical Tools:

Summation:

Let there is sequence of $a_1, a_2, \ldots a_n$

So,

$$a_1 + a_2 + \ldots + a_n = \sum_{i=1}^{n} a_i$$

(pseudo code)

$1+1+\cdots+1$ (n times)

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

2) $\sum_{i=1}^{n} c a_i = c \sum_{i=1}^{n} a_i$

3) $\sum_{i=1}^{n} (a_i + b_i) = \sum_{i=1}^{n} a_i + \sum_{i=1}^{n} b_i$

f

Arithmetic Series:

There should be a common difference in consecutive terms.

- $\sum_{i=1}^{n} i = 1+2+3+\cdots+n$

$$= \frac{n(n+1)}{2} \quad \text{(when } d=1\text{)}$$

$$S_n = \frac{n}{2}\left[2a_1 + (n-1)d\right]$$

Geometric Series:

There is common ratio (r)

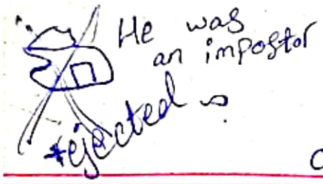
$1+2+4+8+\cdots+n$

$2^0 + 2^1 + 2^2 + \cdots + 2^n = \frac{r^{n+1}-1}{r-1}$

Let series is

$$a_1 + a^2 + a^3 + a^n = \frac{a(r^{n+1}-1)}{r-1} = \frac{a_1(1-r^n)}{1-r}$$

$\boxed{r>1}$ $\boxed{r<1}$



$$\frac{a}{1-r} \quad \boxed{r<1} \qquad \frac{a_1(1-r^n)}{1-r} \quad \boxed{r>1}$$

Harmonic Series:

$$\sum_{i=1}^{n} \frac{1}{i}$$

Quadratic Series:

$$\sum_{i=1}^{n} i^2 = 1+4+9+\dots+n^2$$

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

Selection Sort:

61	40	52	7	17
0	1	2	3	4

Pick minimum & swap i.e.

7	40	52	61	17

7	17	52	61	40

⋮

Similar

Code for this is: Let arr be the array then

```
for(int i=0;i<n;i++)
{
        min=i;
    for(int j=i+1;j<n;j++)
    {
```

```
    if (arr[j] < arr[min])
              min = j
  }
        if (min ! = i)
           swapping ( arr[i], arr[min])
}
```

```
            1          n+1        n
for (i=0; i<n; i+5)                    T(n)=3n+2
       print { "SE" } — n
```

Iteration#	1	2	3	4	k	k
i	0	5	10	15		5k
		↓ 5x1	↓ 5x2	↓ 5x3	↓ (k−1)(5)	

let print (body) run k time

Loop terminate when i=n

$5k = n$

$k = n/5$