Taylor Series e^x

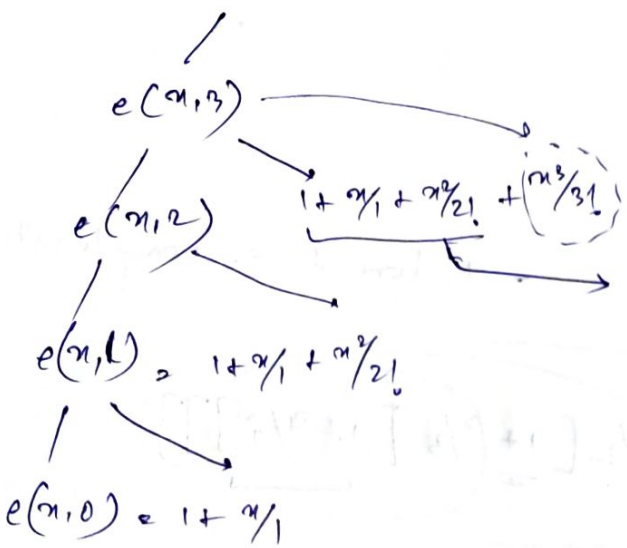
$$e^x = 1 + x/1 + x^2/2! + x^3/3! + x^4/4! + \dots + n \text{ terms}$$

three condition
- ↳ power
- ↳ fact
- ↳ addition.

```
sum(n) = sum(n-1) + n
fact(n) = fact(n-1) * n
pow(x,n) = pow(x,n-1) * x
```

} addition, multiplication are done in returning time.

for eg. e(x,4)

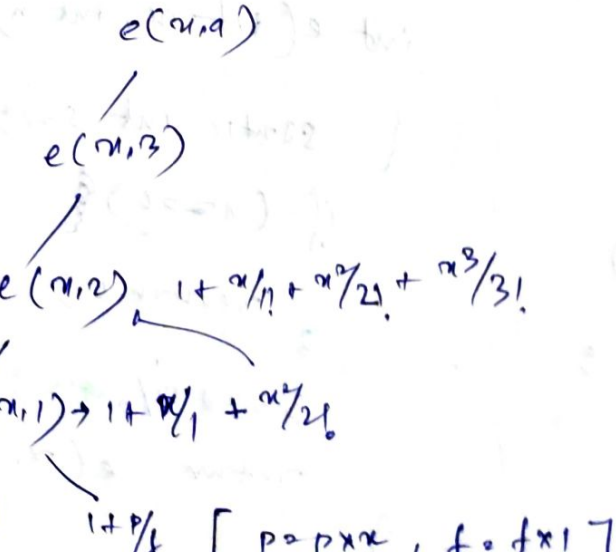


to get x^3 and 3! we need the previous values for which we are using static variables, for storing them.

now, w/ static variables

p f

x*x*x 1*2*3



```
int e(int x, int n)
{
    static int p=1, f=1;
    int r;
    if (n==0)
        return 1
    else{
        r=e(x, n-1);
        p=p*x;
        f=f*n;
        return r+p/f;
    }
}
```

# of multiplication
↳ $n(n+1)$
So, $O(n^2)$
=.

Horner's Rule → low time complexity $O(n)$

$$e^x = 1 + x/1\left[1 + x/2\left[1 + x/3\left[1 + x/4\right]\right]\right]$$

loops

```
int e(int x, int n)
{   int s=1;
    for(; n>0; n--)
    {   s=1+x/n*s;
    }
    return s;
}
```

recursion or calling time

```
int e(int x, int n)
{   static int s=1;
    if (n==0)
        return s;
    s=1+x/n*s;
    return e(x, n-1);
}
```