Assignment - 01

Mehedi Hasan Shahed

21301436

Section: 13

Ans. to the Que. no: 1

$${}^{n}C_{r} + {}^{n}C_{r-1}$$

$$= \frac{n!}{(n-r)!r!} + \frac{n!}{(n-r+1)!(r-1)!}$$

$$= n! \left[\frac{1}{(n-r)!\, r(r-1)!} + \frac{1}{(n-r+1)!\,(r-1)!} \right]$$

$$= \frac{n!}{(r-1)!} \left[\frac{1}{(n-r)!\, r} + \frac{1}{(n-r+1)\,(n-r)!} \right]$$

$$= \frac{n!}{(r-1)!\,(n-r)!} \left[\frac{1}{r} + \frac{1}{(n-r+1)} \right]$$

$$= \frac{n!}{(r-1)!\,(n-r)!} \left[\frac{n+1}{r(n-r+1)} \right]$$

$$= \frac{(n+1)\, n!}{(n+1-r)\,(n-r)!\, r(r-1)!}$$

$$= \frac{(n+1)!}{(n+1-r)!\, r!} = {}^{n+1}C_{r}$$

[Proved]

21301436

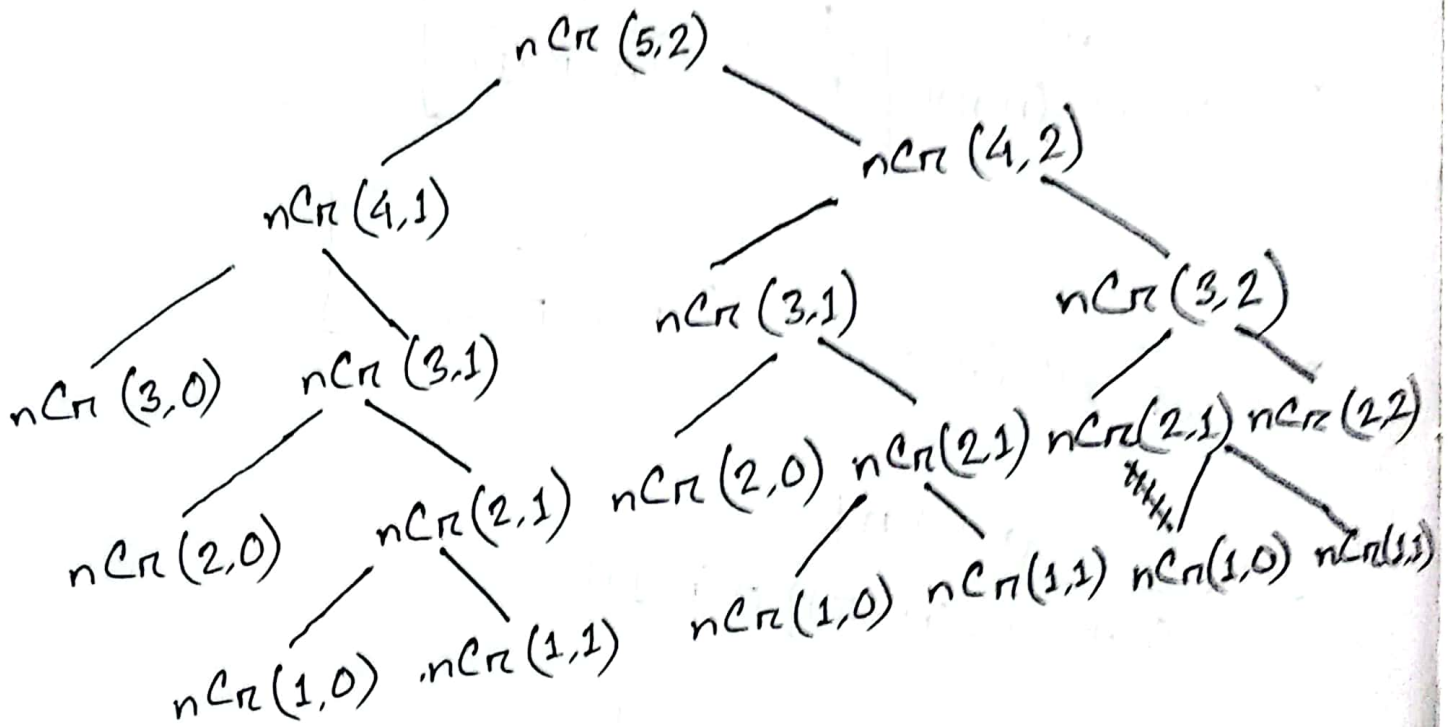
Ans. to the Que. no: 2

```
def nCr(n,r):
    if r>n :
        return 0

    elif r == 0 or r == n:
        return 1

    return nCr(n-1, r-1) + nCr(n-1, r)
```

Ans. to the Que. no: 3



2130143G

Ans. to the Que. no: 4

```
def ter_srch(l, r, key, arr):

    while r >= l:

        mid_1 = l + (r-1) // 3
        mid_2 = r - (r-1) // 3

        if key == arr(mid_1):
            return mid_1
        if key == arr(mid_2):
            return mid_2

        if key < arr[mid_1]:
            r = mid_1 - 1
        elif key > arr[mid_2]:
            l = mid_2 + 1
        else:
            l = mid_1 + 1
            r = mid_2 - 1

    return 'key not found'
```

Ans. to the Que. no: 5

step	No of search
0	n
1	$n/3 = n/3^1$
2	$n/9 = n/3^2$
3	$n/27 = n/3^3$
⋮	⋮
k	$n/3^k$

So, $n/3^k = 1$

$\Rightarrow n = 3^k$

$\Rightarrow \log_3 3^k = \log_3 n$

$\Rightarrow k = \log_3 n$

$\therefore TC = O(\log_3 n)$

2130143G

Ans. to the Que. no: 6

1st loop TC = $O(\log_7 n)$

2nd loop TC = $O(n)$

3rd loop TC = $O(1)$

4th loop TC = $O(n)$

$\therefore$ Total TC = $O(\log_7 n) \times O(n) \times (O(1) + O(n))$

$= O(\log_7 n) \times O(n) \times O(n)$

$= O(n^2 \log_7 n)$

Ans. to the Que. no: 7

1st loop TC = O(n)

The while loop will run untill the value of j reaches $\sqrt{n}$.

∴ While loop TC = $O(\sqrt{n})$

∴ Total TC = $O(n\sqrt{n})$

Ans. to the Que. no: 8

The used loop will be:

```
for i in range(50):
    . . . . .
    . . . . .
```

∴ T.C = O(50)

= ~~O(n)~~ O(1)

21301436

Ans. to the Que. no: 9

Code:

```
count = 0
~~for id in range (0,n):~~
for id in attendance:
    if id % 2 == 0:
        count += 1.
```

Now, if there are n number of students then,

T.C = O(n)

21301436

Ans. to the Que. no: 10

a)
```
def search (arrr):
    if len (arrr) <= 2:
        if arrr [0] > arrr [1]:
            return arrr [0]
        else:
            return arrr [1]
    m = len (arrr) // 2
    if arrr [m+1] <= arrr [m] and arrr [m] >= arrr [m-1]
        return arrr [m]
    elif arrr [m] > arrr [m+1]
        return search (arrr [: m+1])
    else:
        return search (arrr [m:])
```

b) T.C = $O(\log n)$