(4) → i=1, i<=n; i++
j=1 to ~~n~~ j<=n; j++)
arr[i] + arr[j];

T.C = $O(n^2)$

Only(i) → S.C = O(1)

(5) → Sum=0
i=1, i<=n; i++
sum = sum+i

i=2, sum=1+2
i=3, sum=3+3
i=4, sum=6+4

~~[illegible]~~

n.

S.C = i = O(1).

(6) → T.C = O(n).
S.C = n

i=1, i<=n.
arr[i] % 2 == 0
print ——
else
print ——

(7) Recursion not done.

(8) Sum = 0
i = 1

$T.C = O(\log(n))$
$S.C = O(1)$

$i = 2^1, 2^2, 2^3 \ldots 2^k$

$\Rightarrow 2^k \leq n$

(for some value of K 2^k will be less than n)

(9)
```
if n == 0
    return 1

result = 1        i = 1; i <= n; i++
result = result * i
```

$2^k = n$

$k \log 2 = \log n$

$k = \log n$

i = 1	1 × 1
= 2	1 × 2
= 3	1 × 3
= 4	1 × 4
!	! (n)
n	

$T.C = O(n)$
$S.C = O(1)$

(10) Binary Search Algo:

[don't know]

(11) Recursion not done

(12) $(i=1;\ i\le k)$

$(j=1;\ j\le n)$

$T\cdot C = O(nk)$

$S\cdot C = O(1)$

(13). $(i=0;\ i\le n-1)$ $\{j=i+1;\ j<n\}$

i	j
0	$n-0-1 = n-1$
1	$n-(1+1) = n-2$
⋮	⋮
$n-1$	$=(n-1)$

n times

$\therefore T\cdot C = (n-1)\,n.$

$T\cdot C = O(n^2)$

$S\cdot C = O(1)$

(14) [scribbled out]

Recursion not done

(15) (i=1 ; i<=n)
{j=1 ; j<=i}
{ }

i=1 j=1 (1)
i=2 j=1,2 (2)
j=1,2,3 (3)
|
|
i=n j=1,2,3 ... n (n)

∴ 1+2+3+...n

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

$\frac{n^2}{2} + \frac{n}{2}$

T.C = O(n^2)

S.C = O(1)

(16) - result = 0

i=1 ; i<=n
result = result+arr(i)

i=1 ; i<=n
result = result - arr[i]

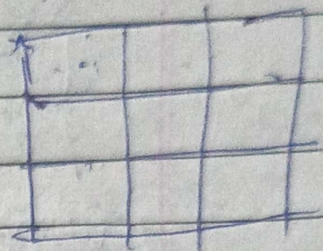
T.C = O(n)
S.C = O(1)

i=1 → 2 operation
i=2 → 2
i=n → 2

∴ TC (2 * n) constant
S.C = O(1)

(17) doubt

(18) matrix 1 → ($m_1 \times n_2$)
matrix 2 → ($n_2 \times n_3$)



(i=1 ; i ≤ n)
{ j=1 ; j ≤ m }

sum=0
k=1 ; k ≤

i=2
j=2 1, 2
k 1, 2, 3

(19) (i = 1 ; i < n)
{j = 1 ; j ≤ i }
{(k = 1 ; k ≤ j)}

$\therefore$ $T \cdot C = n^2 * n^2$
$= O(n^4)$

$S \cdot C =$

$i = n$, $j = \frac{n(n+1)}{2}$

j wrt i

$= O(n^2)$

k wrt j

$k = 1 + 3 + 5 + \dots$

$\frac{n}{2}(2 + (n-1)2)$

$= n/ \cdot n^2 = (n^2)$

(20) n = length of array.
total subsets = 2^n.

(i = 0 ; i ≤ 2^n-1)

{j = 0 ; j ≤ n-1}

~~total oper: 2^n times~~
~~total oper: n times~~

(append arr[j] means j times space increased)

[illegible]
T.C = O(n) * 2^n = O(2^n)
S.C = O(n)

(21) ~~Recursion not done.~~

(22) (i = 1 ; i ≤ n)
{ j = 1 ; j ≤ n }

T.C = O(n * n)
S.C = O(1)

(23) (i = 1 ; i ≤ n)
{ j = 1 ; j ≤ n }

i = 1 , j = 1
i = 2 , 2
i = 3 , 3
i = 4 , 4
⋮
n , n

S.C = O(1)

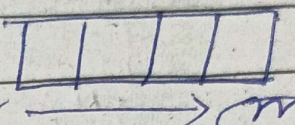
T.C = $\frac{n(n+1)}{2}$ = O(n^2)

(24) $i = 1;\ i \le \frac{n}{2}$

$T.C = \theta\left(\frac{n}{2}\right) = O(n)$

$S.C = O(1)$.

(25) matrix = $n \times m$.

matrix[0] =  $= m$

$\therefore$ $(i = 1;\ i \le n)$

$\{ j = 1;\ j \le m \}$.

$T.C = O(n \times m)$

$S.C = O(n \times m)$

(26) $i = 1;\ i \le n$.

(count = count + 1);

i = 1	count+1
⋮	count+1
i = n	count+1

n times

$\therefore T.C = \theta(n)$

& $S.C = O(1)$

(27) $i=1; i<n$.

$T.C = n$

(28) $n=1=$

$n = 2^k$

:

n

$i=1$

$i=2^k$

$i=4^k$

:

$2^n k$

$2^k = n$

$T.C = O(\log n)$
$S.C = O(1)$

(29) $i=1; i \le n$.

$j=i$

$i=1, j=1$

$i=2, j=2$

$j=3$

:

$j=n$

$1+2+3+\dots+n \quad \frac{n(n+1)}{2}$

$T.C = O(n^2)$
$S.C = O(1)$

(30) (arr, k) n

$i=1; i \le (n-k)$

max value = i

$(j = i+1; j \le i+k)$ & max value = arr(j).

i j

1 2, 3, ... 2+K. } K-times.
2 3, 4, ... 3+K. } K times.
⋮
n-K (n-K+1), ... (n-K+K) } K times

T.C = (n-K) K
S.C = O(1)

BONUS MCQS

(1) len(arr) = n
total Per — = n!

∴ (i=0; i ≤ n!-1) → (n! ~~+1~~) times
= n! times

{(j=n; j ≥ 1; j--)} → n times.

∴ T.C = O(n! * n)
S.C =

doubt

(2) ~~doubt~~ let no. of elements = n
then S.C = O(n)

(3) (4) solved in Q20 in MCQ with process

$S.C = O(n)$ $T.C = O(2^n * n)$

as $2^n >>> n$

$\therefore T.C = O(2^n)$