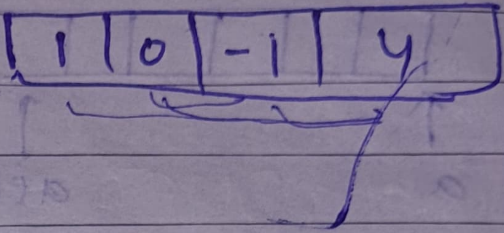
Page No.
Date :

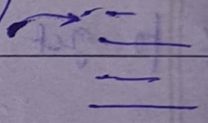
Arrays

→ Generating subarrays → continous



```
for(int i=0; i<n; i++)
 {
     for(int j=i ; j<n; j++)
     {
         for(int k=i ; k<=j; k++)
         {
             cout<<a[k];
         }
         cout<<"\n";
     }
 }
```

One line



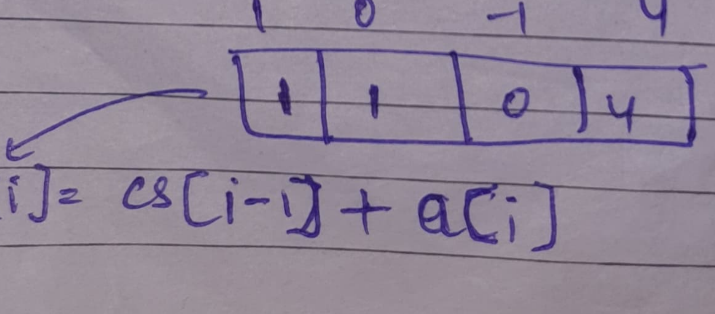
→ Maximum Subarrays sum

1st way bruteforce :→ add in above loop ↗ sum, temp.

$O(n^3)$

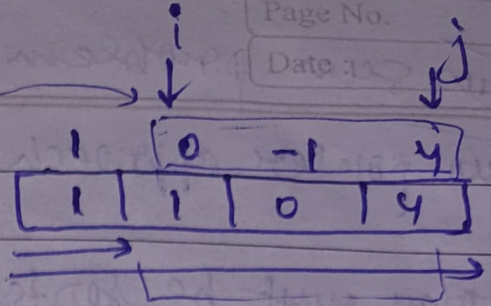
2nd way cumulative sum :→ 1st make a cumulative sum array

$O(n^2)$



cs[i] = cs[i-1] + a[i]

~~[illegible]~~ also for



sum from i to j

$Sum = cs[j] - cs[i-1]$

3rd way: Kadane's Algo :→

1st maintain 2 variable curr and max Till now
initialize, curr = 0
Max Till now = 0

2nd Iterate from i=0 to i=n-1, and do

```
curr ~~+~~= curr + a[i]
Max Till now = max [Max Till now, curr]
```

but if curr is curr>0 then only

otherwise,

3rd if curr<0, then do curr = 0 ;
maxTillnow = max(maxTillNow, a[i]) {usse element ko le hi mat, phir se array shuru kar}

4th when loop ends, output max till now.

Two sum problem !!

→ Two pointer approach

(1) array must be sorted.

(2) if (a[i] + a[j] > sum) $\{\overset{i \rightarrow}{1}, 3, 5, 7, 10, 11, 12, \overset{\leftarrow j}{13}\}$.

j--; → toh its kam karna h matlab

else if (a[i] + a[j] < sum) → toh its badhana h matlab.

i++;