

Q1 arr: 10 -2 8 15 6 -8 12 4 -1  
0 1 2 3 4 5 6 7 8

Q: i j  $\sum_{k=i}^j arr[k]$   
2 4 29  
5 8 12  
0 2 16

Q2 given array of size N and queries. For every find subarray sum from i to j

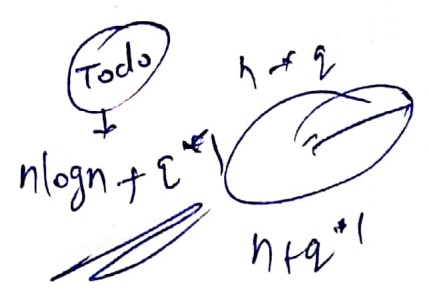
```
1) int sum(int arr[], int i, int j) {    O(n-j-i)
    int sum = 0;
    for (int k = i; k <= j; k++) {
        sum += arr[k];
    }
    return sum;
}
```

2) Find cumulative array of arr  
c-arr = 10 8 16 31 37 29 46 50 49  
precompute (N+O(1)) query

```
int sum(int arr[], int i, int j, int q) {
    int cum-sum[1];
    while (q-- > 0) {
        int i = sc.nextInt();
        int j = sc.nextInt();
        int
```

```
int c-sum[] = new int[n]; c-sum[0] = arr[0];
for (i = 1; i < n; i++) {
    c-sum[i] = arr[i] + c-sum[i-1];
}
while (q-- > 0) {
    ans = solve(i, j, arr, c-sum);
    print(ans);
}
```

```
int solve ( ) {
    if (i == 0) return c-sum[j];
    return c-sum[j] - c-sum[i-1];
}
```



// computing suffix:

sub[n] = 1;

sub[n-1] = arr[n-1];

for (int i = n-2; i >= 0; i--) {

sub[i] = sub[i+1] + arr[i];

}

Q

A = a b b x y a y b b a  
 B = y a b b a x y a

$a[i \text{ to } j] == b[k \text{ to } l] \rightarrow \text{True.}$   
 $!= \rightarrow \text{False.}$

Q =	i	j	k	l	o/p
	0	2	1	3	T
	4	6	0	2	F
	7	9	2	4	T
	2	3	5	7	F

$$1) O(Q \times N) \rightarrow \sum_{i=0}^N \text{if}(A[i] != B[i]) \rightarrow T$$

2) prefix array.

PHA: a a+b a+b+b . . . .

PHB: y y+a y+a+b . . . .

if ( PHA[j] - ((i!=0)? PHA[i-1]: 0) ==  
 PHB[l] - (k!=0)? PHB[l-1]: 0) } Hash string  
 collision  
 return True;

else  
 return False;

To handle

→ compute power array  $O(N)$

→ PHA } compute  $O(N)$   
 PHB

→ HA compare with HB  $O(i)$

→ calculate  $d \rightarrow (k-i) \rightarrow (k-i) \cdot O(1)$   
 $(k > i) ? HA \times p^d : HB \times p^d$  } # modulo arithmetic  
 where necessary

→ return true.

$$\rightarrow \boxed{\max(N, M) + N + M + O(1)}$$

2CS

$$HA = [1:3] : [ap^2 + bp^3 + cp^4] \quad ; \quad PHA[3] = PHA[0]$$

$$HB = [2:4] : [ap^3 + bp^4 + cp^5] \quad ; \quad PHB[4] = PHB[1]$$

code:

$$p[0] = p \quad (p \text{ is variable}) \quad \{ \text{if doesn't work increase } p \}$$

$$\forall_{i=1}^{m \times (N/M)} p[i] = (p[i-1] \times p) \% \text{mod}$$

$$\text{int } PHA[N] = \{0\};$$

$$PHA[0] = A[0] \times p[0]$$

$$\forall_{i=1}^N PHA[i] = (PHA[i-1] + A[i] \times p[i] \% \text{mod}) \% \text{mod}$$

$$\text{int } PHB[M];$$

$$PHB[0] = B[0] \times p[0]$$

$$\forall_{i=1}^M PHB[i] = (PHB[i-1] + B[i] \times p[i] \% \text{mod}) \% \text{mod}$$

loop(Q)

Read(i, j, k, l)

$$HA = PHA[j] - ((i \neq 0) ? PHA[i-1] : 0)$$

$$HB = PHB[l] - ((k \neq 0) ? PHB[k-1] : 0)$$

$$d = \text{abs}(k-i)$$

$$i \neq (k > i)$$

$$HA = (HA \times p[d-1]) \% \text{mod}$$

else

$$HB = (HB \times p[d-1]) \% \text{mod}$$

} equal powers

$$\text{if } (HA == HB)$$

return True;

else

return False;