Lecture 4: Prefix sum

⑥ Given an array of N elements, find sum of all elements from index "s" to "e"

Brute Force

```
sum = 0
for i in range (s, e+1):
    sum += A[i]

return sum
```

TC: O(N)

SC: O(1)

But if there are multiple pairs of (s,e)

TC: O(N * Q) SC: O(1)

Prefix sum: sum of all elements starting from s to e

```
ps = [0]*N
ps[0] = A[0]
for i in range(1,N):
    ps[i] = ps[i-1] + A[i]
```

sum from index 4 to 8

ps[8] − ps[3]

sum from index 0 to 6

ps[6]

(7) Find the equilibrium index of an array

↳ sum on left == sum on right

```
for i in range(1, n):
    if ps[i-1] == ps[n-1] - ps[i]
        return True
return False
```

(8) Find the count of special indices in an array if that element is removed,

sum of even-indexes = sum of odd-indexes

```
N = len(A)
even_PS = [0 for i in range(N)]
odd_PS = [0 for i in range(N)]

even_PS[0] = A[0]
odd_PS[0] = 0

for i in range(1, N):
    if i%2 == 0:
        even_PS[i] = even_PS[i-1] + A[i]
        odd_PS[i] = odd_PS[i-1]
    else
        odd_PS[i] = odd_PS[i-1] + A[i]
        even_PS[i] = even_PS[i-1]

count = 0
for i in range(N):
    if i == 0:
        even_sum = odd_PS[N-1]
        odd_sum = even_PS[N-1] - even_PS[0]
    else:
        even_sum = even_PS[i-1] + odd_PS[N-1] - odd_PS[i]
        odd_sum = odd_PS[i-1] + even_PS[N-1] - even_PS[i]

    if even_sum == odd_sum:
        count += 1
return count
```

⑨ Pick from both sides!

Given an array of A of N elements, select B elements; some (all) from left / right to get maximum sum

```
"get prefix sum array"
if B == N
    return ps[N-1]
max-sum-possible = ps[N-1] - ps[N-B-1]
for i in range(B):
    prospect = ps[i] + ps[N-1] - ps[N-B+i]
    if prospect > max-sum-possible:
        max-sum-possible = prospect
return max-sum-possible
```