2h 5m left

## 2. Stable Segments

In an organization, there are *n* servers, each with a capacity of *capacity[i]*. A contiguous subsegment *[l, r]* of servers is said to be stable if *capacity[l] = capacity[r] = sum[l+1, r-1]*. In other words, the capacity of the servers at the endpoints of the segment should be equal to the sum of the capacities of all the interior servers.

Find the number of stable sub-segments of length 3 or more.

**Example**

For example, *n=5* and *capacity=[9, 3, 3, 3, 9]*.

| Segment | First & Last Capacity | Interior Capacity Sum | Balanced |
|---|---|---|---|
| [9, 3, 3] | 9 | 6 | No |
| [3, 3, 3] | 3 | 3 | Yes |
| [3, 3, 9] | 3 | 3 | No |
| [9, 3, 3, 3] | 9 | 6 | No |
| [3, 3, 3, 9] | 3 | 9 | No |
| [9, 3, 3, 3, 9] | 9 | 9 | Yes |

There are 2 stable subsegments: [3, 3, 3] and [9, 3, 3, 3, 9]. Return 2.

**Function Description**

Complete the function *countStableSegments* in the editor below.

*countStableSegments* has the following parameter:

Language C++20

Autocomplete Ready

```
#include <bits/stdc++
/*
 * Complete the 'co
below.
 *
 * The function is
 * The function a
parameter.
 */

int countStable

}

int main() ...
```

Test Resu

2h 4m left

ALL

There are 2 stable subsegments: [3, 3, 3] and [9, 3, 3, 3, 9]. Return 2.

**Function Description**

Complete the function *countStableSegments* in the editor below.

*countStableSegments* has the following parameter:

*int capacity[n]:* the capacities of each server

**Returns**

*int:* the number of stable segments

**Constraints**

- $1 \le n \le 3 * 10^5$
- $1 \le capacity[i] \le 10^9$

▶ **Input Format For Custom Testing**

▼ **Sample Case 0**

**Sample Input For Custom Testing**

```
STDIN          FUNCTION
-----          --------
7        →     n = 7
9        →     capacity = [9, 3, 1, 2, 3, 9, 10]
3
1
2
3
9
10
```

Language C++20

Autocomplete Ready

```
1  #include <bits/stdc++.h>...
9
10 /*
11  * Complete the 'countS
   below.
12  *
13  * The function is exp
14  * The function accep
   parameter.
15  */
16
17 int countStableSeg
18
19 }
20
21 int main()...
```

Test Resul

2h 4m left

ALL

### Input Format For Custom Testing

The first line contains an integer, *n*, the number of elements in *capacity*.

Each line *i* of the *n* subsequent lines (where $0 \leq i < n$) contains an integer, *capacity[i]*.

### Sample Case 0

#### Sample Input For Custom Testing

```
STDIN       FUNCTION
-----       --------
7       →   n = 7
9       →   capacity = [9, 3, 1, 2, 3, 9, 10]
3
1
2
3
9
10
```

#### Sample Output

```
2
```

#### Explanation

*The stable segments are* [9, 3, 1, 2, 3, 9] and [3, 1, 2, 3].

### Sample Case 1

#### Sample Input For Custom Testing

```
STDIN       FUNCTION
-----       --------
5       →   n = 5
6       →   capacity = [6, 1, 2, 3, 6]
```

Language C++20

Autocomplete Ready

```
1  #include <bits/st
9
10 /*
11  * Complete the
    below.
12  *
13  * The functio
14  * The functi
    parameter.
15  */
16
17 int countSt
18
19 }
20
21 int main(
```

Test

**Sample Output**

```
2
```

**Explanation**

The stable segments are [9, 3, 1, 2, 3, 9] and [3, 1, 2, 3].

## Sample Case 1

**Sample Input For Custom Testing**

```
STDIN       FUNCTION
-----       --------
5       →   n = 5
6       →   capacity = [6, 1, 2, 3, 6]
1
2
3
6
```

**Sample Output**

```
1
```

**Explanation**

*The entire array is a stable segment.*

```
 * Complete
below.
 *
 * The func
 * The func
parameter.
 */

int counts

}

int main
```

Test Res