a : 3 1 1 1 5 1 2 1 : 8

t = 3

{3} {1,1,1} 1 + 3 = 4

{3} {1,2} 1 + 2 = 3 ✓

{3} {2,1} 1 + 2 = 3 ✓

{1,1,1} {2 1} 3 + 2 = 5

{1,1,1} {1,2} 3 + 2 = 5

$l_1$

$l_2$

$l_1 + l_2$

Constraint

1. Two subarray each with sum = target
2. These subarray should not overlap
3. Sum of length of these sub array should be minimum
4. 1<= a[i] <= 100000

1) Brute force

a) find all subarray with sum = target — $O(n^2)$

b) choose two subarry which are not overlapping and having min length — $O(n^4)$

(a) ⇒ 1 2 3 4 .... (len = n)

subarray starting with 1 ⇒ {1}, {1,2}, {1,2,3} ..... n-1

2 ⇒ {2}, {2,3}, {2,3,4} ..... n-2

⋮ ⋮ ⋮

1

Total = 1+2+3+4+5+... n-1 ⇒ $\frac{n(n-1)}{2} = O(n^2)$

(b) ⇒ we have $n^2$ nos. out of which we have to choose 2 nos.

for ( i = 1 to $n^2$ ) ⇒ (1)

for (j = 1 to $n^2$ ) ⇒ (2)

i & j

$O(n^4)$ ✓

2) optimise approach (Dynamic Programming)

a : 3 1 1 1 5 1 2 1 : 8

len[i] ⇒ length of subarry with sum = t starting with i

t = 3 min[i] ⇒ min length of subarry with sum = t in array [i, .... n-1]

compute len[i]

| | 0 | 1 | 2 | 3 | | | | |
|---|---|---|---|---|---|---|---|---|
| a : | 3 | 1 | 1 | 1 | 5 | 1 | 2 | 1 |
| len : | 1 | 3 | ∞ | ∞ | ∞ | 2 | 2 | ∞ |

(i, i+1 ... j)

(5) 2 (i (1) j)

(i, 1, 1) ⇒ 3

i = 0, j = 0, sum = 0, t = 3

for (j = 0 to n-1)

sum += a[j]

while (sum > t)

sum -= a[i]

i++

if (sum == target)

$(i, 1, j, 1) \Rightarrow 3$

if (sum == target)
len[i] = j - i + 1
3 - 1 + 1
$\Rightarrow 3$

Compute min[i]

| len: | 1 | 3 | ∞ | ∞ | ∞ | 2 | 2 | ∞ |
|---|---|---|---|---|---|---|---|---|

| min: | 1 | 2 | 2 | 2 | 2 | 2 | 2 | ∞ |
|---|---|---|---|---|---|---|---|---|

R → L
min = ∞ 2 1

Compute ans:

i : 0 → n-1

len[i] + min[i + len[i]]

| a: | 3 | 1 | 1 | 5 | 1 | 2 | 1 | |
|---|---|---|---|---|---|---|---|---|

| len: | 1 | 3 | ∞ | ∞ | ∞ | 2 | 2 | ∞ |
|---|---|---|---|---|---|---|---|---|

L → R
R → L
R → L

| min: | 1 | 2 | 2 | 2 | 2 | 2 | 2 | ∞ |
|---|---|---|---|---|---|---|---|---|

len[i] ⇒ O(n)
min[i] ⇒ O(n)
O(n)

O(n)

(1 + 2) = 3
(3 + 2) = 5

3) reduce space

use only min by computing length from R to L

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| a: | 3 | 1 | 1 | 1 | 5 | 1 | 2 | 1 |
| min: | ∞ | 2 | ∞ | 2 | 2 | 2 | 2 | ∞ |

i, j, = n-1,
m = ∞ 2
res = ∞ 3
sum = 0

sum = 0 1 2 3 2

[2, 1]
[1, 2]
[1, 1]
3
3

[j+1, n-1]
3, (5 1 2 1)
2 = 5

sum = 0 1 2 3 5 R
3

3 + 2 =
2 = 3

3, (1, 2)
res = 3