

Continuous Subarray Sum.

find multiple of k

$[2, 3, 2, 4, 6, 7]$, $k = 6$.

$$\Rightarrow S \% k == 0.$$

prefix
sum 23 25 29 35 42.

$$\text{Subarray sum}(i, j) = S[j] - S[i-1]$$

$$\downarrow \qquad \downarrow$$
$$ak + r_1 \qquad bk + r_2.$$

$$\Rightarrow (a-b)k + (r_1 - r_2)$$

for multiple of k .

$$r_1 - r_2 == 0$$

$$\Rightarrow [2, 3, 2, 4, 6, 7]$$

prefix
sum 23 25 29 35 42.

$$r \qquad 5 \quad 1 \quad 5 \quad 5 \quad 0.$$
$$\hat{i} \qquad \qquad \hat{j}$$

if prefix sums i, j have the same,
subarray $(j+1, i)$ is multiple of k .

const fn = (nums, k) => {

const map = { 0: -1 };

let sum = 0;

for (let i = 0; i < nums.length; i++) {

sum += nums[i];

const r = sum % k;

if (r in map) {

const j = map[r];

if (i - j >= 2) {

return true;

}

} else {

map[r] = i;

}

}

return false;

}

special cases.

$k \leq 0$.

[2, 3, 2, 4, 6, 7], $k = 6$.

const fn = (nums, k) => {

for (let i = 0; i < nums.length; i++) {

if (nums[i] === 0 &&

nums[i+1] === 0) {

return true;

}

}

if (k === 0) {

return false;

}

k = Math.abs(k);

const map = { 0: -1 };

let sum = 0;

for (let i = 0; i < nums.length; i++) {

sum += nums[i];

const r = sum % k;

if (r in map) {

const j = map[r];

if (i - j >= 2) {

return true;

}

$k < 0$:

works the same with $k > 0$.

So just set $k = |k|$.

$k === 0$:

[3, 4, 5, 6, 7], $k = 0$.

[3, 4, 0, 0, 1], $k = 0$.

∵ nums is a list of non-negative integers.

∴ only if subarray sum is '0' could be multiple of '0'.

⇒ looking for

consecutive zeros

also '0' is multiple of any number

```
} else {  
    map[r] = i;  
    }.
```

```
}  
return false;
```

```
}
```