Minimum number of swaps
for bracket balancing

Example:

]][[⟹ Find number of minimum swaps for bracket balancing.

We can solve this in

> Time Complexity: $O(n)$
> Space Complexity: $O(n)$

Algorithm

Iterate through char's of string

```
if(S[i] == '[') {
    count++
    idx++
} else {
    count--
    if(count < 0) {
        ans += v[idx] - i
        swap(S[i], S[v[idx]])
        count = 1
        idx++
    }
}
```

array V consists the indexes of '[' because while we are iterating through the bracket string and if we find any unbalanced ']' bracket, we use this array and get the corresponding '['.

> count = 0 ⟹ Balanced
> count < 0 ⟹ unbalanced by following] and next [

> idx is pointer for open bracket indexes which we stored in array V

Here we are doing ans += v[idx] - i, what it means we are swapping all the brackets b/w i & v[idx] and finally swap i & v[idx]

Application

]][[

V = [2, 3]

Iterate through char's of string

i=0 S[i] =]
count = 0
} ⟹ ans = v[0] - 0
ans = 2 - 0 = 2
swap 0th ⟺ 2nd

[[ ] [count = 1
           idx = 1

---

i=1 S[i] =]
count = 1
idx = 1
} count = 0
no swapping bec brackets are balanced until i=1

[ ]] [

---

i=2 S[i] =]
count = 0
idx = 1
ans = 2
} count = -1
unbalanced
ans += v[idx] - i
ans += v[1] - 2
ans += 3 - 2 = 1
ans += 1 ⟹ 3
swap 2nd ⟺ 3rd

[ ]] [⟹ [ ] [ ]

count = 1

---

i=3 count = 1
S[i] =]
ans = 3
} count = 0
Brackets Balanced