

Question 1

SegregateFromBeginning(arr)

```
n = len(arr)
swaps = 0
for i = 0 to n:
    if arr[i] % 2 == 0:
        for j = i + 1 to n:
            if arr[j] % 2 == 1:
                arr[i] , arr[j] = arr[j] , arr[i]
                swaps++
                break
return swaps
```

SegregateFromEnd(arr)

```
n = len(arr)
swaps = 0
for i = n-1 to -1:
    if arr[i] % 2 == 0:
        for j = i - 1 to -1:
            if arr[j] % 2 == 1:
                arr[i] , arr[j] = arr[j] , arr[i]
                swaps++
                break
return swaps
```

print(min(SegregateFromBeginning(arr), SegregateFromEnd(arr)))

Time Complexity: $O(n^2)$

Auxiliary Space: $O(1)$

در این مسئله، تابع اول تعداد جابجایی‌ها را برای حالتی محاسبه می‌کند که اعداد زوج در انتهای لیست باشند و لیست را از اول به آخر پیمایش می‌کنیم.

تابع دوم نیز تعداد جابجایی‌ها را برای حالتی محاسبه می‌کند که اعداد زوج در ابتدای لیست باشند و لیست را از آخر به اول پیمایش می‌کنیم.