Solutions

August 15, 2019

1 Solutions

1. The long way to solve it

[1]:

```
def sort_array(source_array):
    if len(source_array)==0:
        return []
    else:
        for i in range(len(source_array)):
            for j in range(len(source_array)):
                if source_array[i]%2!=0 and source_array[j]%2!=0 and␣
 ↪source_array[i]<source_array[j]:
                    temporal=source_array[i]
                    source_array[i]=source_array[j]
                    source_array[j]=temporal
        return source_array
```

to see that this works, we can test it for 10 random tests

[3]:

```
import random
a=list(range(100))
for num_test in range(10):
    b=random.sample(a,10) # we take 10 random numbers
    c=b.copy() # we generate a copy of the list to dont change its values
    print("before",b)
    print("After",sort_array(c))
    print("-"*10)
```

```
before [47, 23, 12, 64, 66, 54, 33, 26, 73, 11]
After [11, 23, 12, 64, 66, 54, 33, 26, 47, 73]
----------
before [83, 82, 31, 72, 10, 36, 56, 76, 60, 27]
After [27, 82, 31, 72, 10, 36, 56, 76, 60, 83]
----------
before [30, 56, 60, 98, 75, 50, 46, 55, 58, 18]
After [30, 56, 60, 98, 55, 50, 46, 75, 58, 18]
----------
before [71, 50, 60, 12, 88, 48, 55, 68, 31, 85]
```

```
After [31, 50, 60, 12, 88, 48, 55, 68, 71, 85]
----------
before [89, 73, 27, 4, 60, 87, 75, 31, 32, 91]
After [27, 31, 73, 4, 60, 75, 87, 89, 32, 91]
----------
before [17, 19, 90, 92, 65, 1, 67, 46, 34, 87]
After [1, 17, 90, 92, 19, 65, 67, 46, 34, 87]
----------
before [47, 94, 20, 17, 28, 51, 77, 45, 97, 92]
After [17, 94, 20, 45, 28, 47, 51, 77, 97, 92]
----------
before [91, 24, 89, 39, 41, 26, 70, 80, 46, 0]
After [39, 24, 41, 89, 91, 26, 70, 80, 46, 0]
----------
before [57, 1, 8, 2, 73, 67, 31, 87, 30, 90]
After [1, 31, 8, 2, 57, 67, 73, 87, 30, 90]
----------
before [24, 19, 46, 53, 68, 1, 10, 30, 41, 91]
After [24, 1, 46, 19, 68, 41, 10, 30, 53, 91]
----------
```

So we can see this works, how ever there's a shorter version of doing this and this is done in a way python allow us to work with lists.

[4]:
```
def sort_array(arr):
    odds = sorted((x for x in arr if x%2 != 0), reverse=True)
    return [x if x%2==0 else odds.pop() for x in arr]
```

2. To make the second one we can make use of the first one since it is quite similar but this time setting an additional condition over the even numbers.

[5]:
```
def sort_array(a):
    if len(a)==0:
        return []
    else:
        for i in range(len(a)):
            for j in range(len(a)):
                if a[i]%2!=0 and a[j]%2!=0 and a[i]<a[j]:
                    temporal=a[i]
                    a[i]=a[j]
                    a[j]=temporal
                elif a[i]%2==0 and a[j]%2==0 and a[i]>a[j]:
                    temporal=a[i]
                    a[i]=a[j]
                    a[j]=temporal
        return a
```

Again we test it to check whether it works or not

[6]:

```python
a=list(range(100))
for num_test in range(10):
    b=random.sample(a,10) # we take 10 random numbers
    c=b.copy() # we generate a copy of the list to dont change its values
    print("before",b)
    print("After",sort_array(c))
    print("-"*10)
```

```
before [63, 13, 89, 40, 92, 34, 7, 79, 90, 56]
After [7, 13, 63, 92, 90, 56, 79, 89, 40, 34]
----------
before [99, 68, 69, 20, 62, 55, 31, 57, 67, 45]
After [31, 68, 45, 62, 20, 55, 57, 67, 69, 99]
----------
before [14, 77, 57, 39, 93, 19, 34, 17, 59, 76]
After [76, 17, 19, 39, 57, 59, 34, 77, 93, 14]
----------
before [90, 70, 93, 21, 92, 43, 79, 5, 66, 61]
After [92, 90, 5, 21, 70, 43, 61, 79, 66, 93]
----------
before [68, 57, 81, 9, 2, 75, 38, 95, 86, 27]
After [86, 9, 27, 57, 68, 75, 38, 81, 2, 95]
----------
before [54, 17, 76, 57, 15, 11, 80, 89, 7, 82]
After [82, 7, 80, 11, 15, 17, 76, 57, 89, 54]
----------
before [5, 46, 66, 40, 78, 95, 49, 88, 26, 23]
After [5, 88, 78, 66, 46, 23, 49, 40, 26, 95]
----------
before [96, 16, 87, 5, 64, 34, 68, 31, 47, 46]
After [96, 68, 5, 31, 64, 46, 34, 47, 87, 16]
----------
before [74, 23, 18, 6, 45, 27, 60, 47, 86, 71]
After [86, 23, 74, 60, 27, 45, 18, 47, 6, 71]
----------
before [21, 88, 0, 74, 4, 39, 9, 91, 2, 19]
After [9, 88, 74, 4, 2, 19, 21, 39, 0, 91]
----------
```

so we see that it works but as always the shorter version of this can be done by

[7]:

```python
def sort_array(xs):
    es = sorted(x for x in xs if x % 2 == 0)
    os = sorted((x for x in xs if x % 2 != 0), reverse=True)
    return [(es if x % 2 == 0 else os).pop() for x in xs]
```

3. Here we are going to use the method `sorted` just to make it shorter but since in the two last methods we have implemented a method to order it shouldn't take lot of time to implement your own function to sort.

An important thing here is that the number of odds has to be the same number of evens to perform this operation and to guarantee that we make a while loop.

[11]:

```
ready=True
while ready:
    a=list(range(100))
    b=random.sample(a,20)
    c=b[10:]
    b=b[:10]
    odds=[]
    even=[]
    for i in b:
        if i%2!=0:
            odds.append(i)
        else:
            even.append(i)
    for j in c:
        if j%2!=0:
            odds.append(j)
        else:
            even.append(j)
    odds=sorted(odds)
    even=sorted(even,reverse=True)
    if len(odds) == len(even):
        ready=False
print("before",c,b)
result=[0]*20
aux_odds=0
aux_evens=0
for i in range(len(result)):
    if i%2!=0:
        result[i]=odds[aux_odds]
        aux_odds+=1
    else:
        result[i]=even[aux_evens]
        aux_evens+=1
print("-"*30)
print("After",result)
```

```
before [5, 68, 38, 83, 48, 1, 85, 10, 37, 32] [74, 9, 35, 20, 56, 27, 2, 75, 94,
65]
------------------------------
After [94, 1, 74, 5, 68, 9, 56, 27, 48, 35, 38, 37, 32, 65, 20, 75, 10, 83, 2,
85]
```

4. This part can be easily done if did the last part, since we can use the function set and then transform that into a list again

```
list(set(result))
```