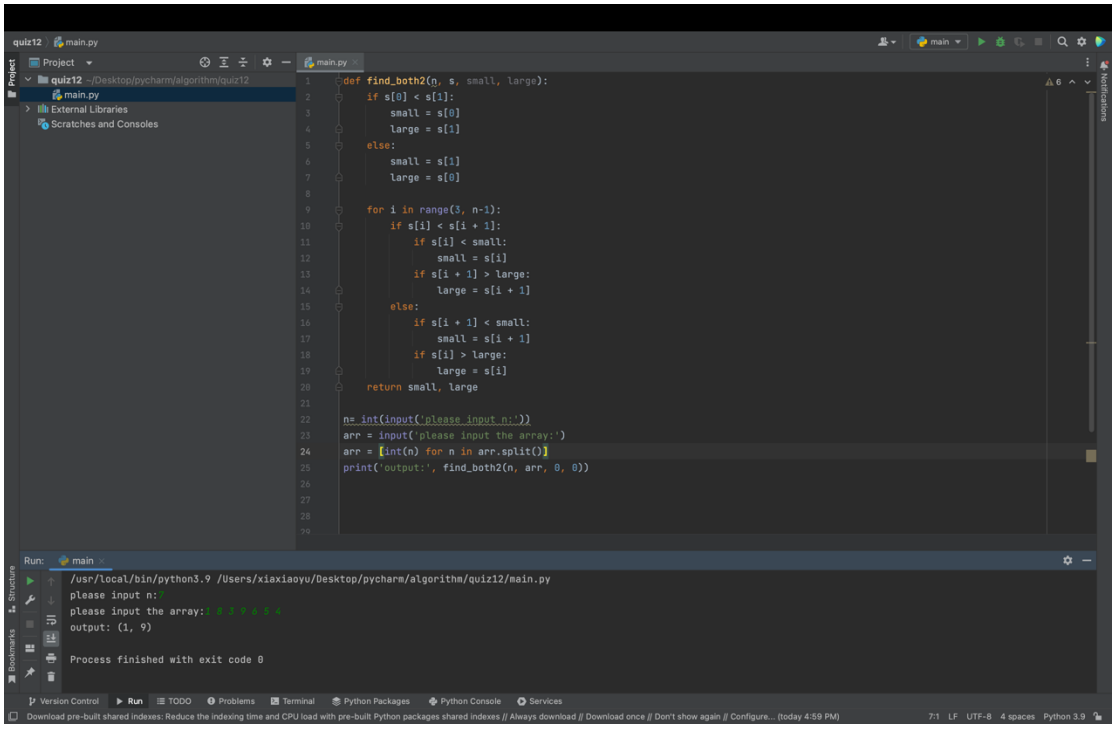
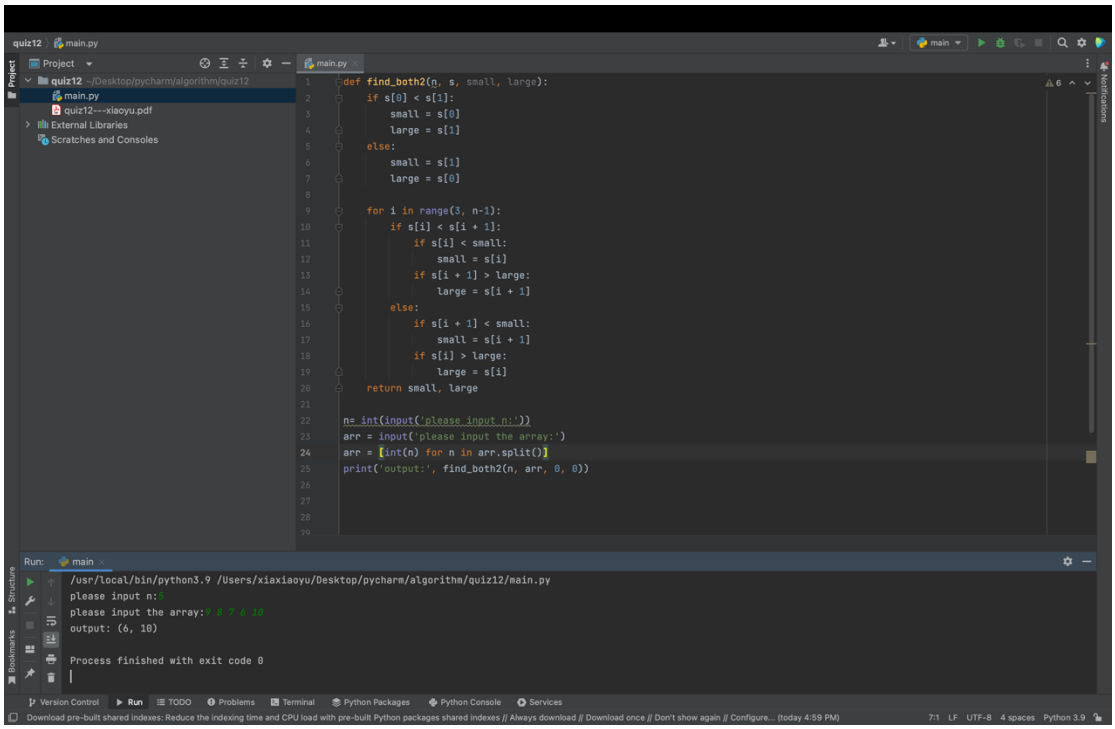
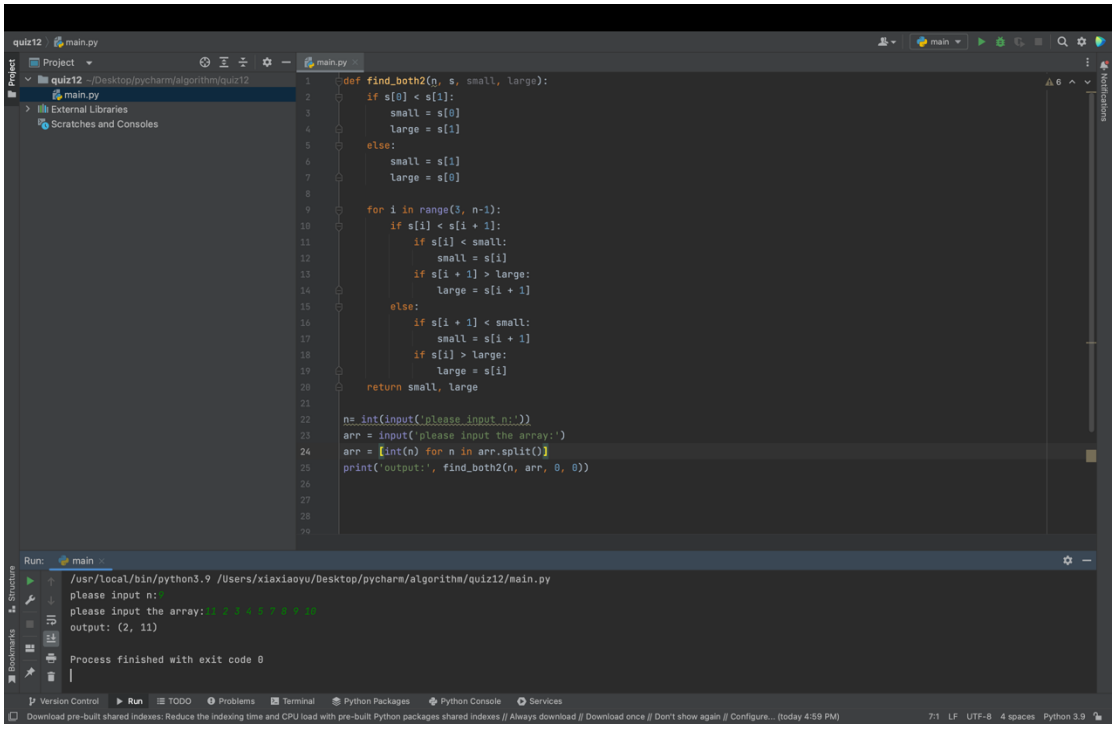
Test1


quiz12 main.py
Project
quiz12 ~/Desktop/pycharm/algorithm/quiz12
main.py
External Libraries
Scratches and Consoles
def find_both2(n, s, small, large):
if s[0] < s[1]:
small = s[0]
large = s[1]
else:
small = s[1]
large = s[0]
for i in range(3, n-1):
if s[i] < s[i + 1]:
if s[i] < small:
small = s[i]
if s[i + 1] > large:
large = s[i + 1]
else:
if s[i + 1] < small:
small = s[i + 1]
if s[i] > large:
large = s[i]
return small, large
n= int(input('please input n:'))
arr = input('please input the array:')
arr = [int(n) for n in arr.split()]
print('output:', find_both2(n, arr, 0, 0))
Run: main
/usr/local/bin/python3.9 /Users/xiaxiaoyu/Desktop/pycharm/algorithm/quiz12/main.py
please input n:7
please input the array:3 8 3 9 6 5 4
output: (1, 9)
Process finished with exit code 0

Test2


quiz12 main.py
Project
quiz12 ~/Desktop/pycharm/algorithm/quiz12
main.py
quiz12---xiaoyu.pdf
External Libraries
Scratches and Consoles
def find_both2(n, s, small, large):
if s[0] < s[1]:
small = s[0]
large = s[1]
else:
small = s[1]
large = s[0]
for i in range(3, n-1):
if s[i] < s[i + 1]:
if s[i] < small:
small = s[i]
if s[i + 1] > large:
large = s[i + 1]
else:
if s[i + 1] < small:
small = s[i + 1]
if s[i] > large:
large = s[i]
return small, large
n= int(input('please input n:'))
arr = input('please input the array:')
arr = [int(n) for n in arr.split()]
print('output:', find_both2(n, arr, 0, 0))
Run: main
/usr/local/bin/python3.9 /Users/xiaxiaoyu/Desktop/pycharm/algorithm/quiz12/main.py
please input n:5
please input the array:3 8 7 6 10
output: (6, 10)
Process finished with exit code 0

Test3



Time complexity:

Time complexity:

$\frac{3n}{2}-2$ if n is even

$\frac{3n}{2}-\frac{3}{2}$ if n is odd

even:

① each two boys need one comparison

so total we need $\frac{n}{2}$ comparisons.

Then we get $2\cdot\frac{n}{2}$ information.

② we must do at least $2n-2-n=n-2$ comparisons

because adversary provides us just one information.

③ $\frac{n}{2}+n-2=\frac{3n}{2}-2$

adversary need the algorithm to do additional comparisons.

So Time complexity is $\frac{3n}{2}-2$ if n is even.

odd:

① make one comparison between s[0] and s[1] to get initial 'small' and 'large'

② we need three types of comparison

(1) the neighbor element

(2) if small < min

(3) if large > max.

in each type, we have n-3 elements to compare,

so, the number of comparisons for this part is $3 \cdot \frac{n-3}{2}$

③ make the last comparison with both min and max. $\Rightarrow 2$

so total time complexity is $1 + 3 \cdot \frac{n-3}{2} + 2 = \frac{3n}{2} - \frac{3}{2}$

So Time complexity is $\frac{3n}{2} - \frac{3}{2}$ if n is odd.