

✓ Points to remember

- Codes asked are relatively easier in comparison to SDE roles
- Pythonic syntax is preferred in most companies
- Focus is on optimized code
- Start with the simplest solution and then improve
- A lot of questions are repeated so mugging up the approaches help
- Build intuition using Python Tutor

1. Find the kth largest/smallest item from a list

```
L = [12,23,1,4,56,34,22,3]
k=3
```

```
L.sort(reverse=True)
print(L[k-1])
```

 23

2. Check if an array is sorted

```
L = [1,2,3,4,5]
```

```
flag = True
```

```
for i in range(0, len(L)-1):
    if L[i] > L[i+1]:
        flag = False
```

```
if flag:
    print('sorted')
else:
    print('not sorted')
```

```
sorted
```

3. Find Min/Max in a given array

```
L = [21,1,34,23,54,11,10]
```

```
max_val = L[0]
```

```
for i in L:
    if i > max_val:
        max_val = i
```

```
print(max_val)
```

```
54
```

4. Find the first element to occur k times in an array

```
L = [1,1,2,3,4,4,5,5]
```

```
k = 2
```

```
d = {}
```

```
for i in L:
    if i in d:
        d[i] = d[i] + 1
    else:
        d[i] = 1
```

```
for i in d:
    if d[i] == k:
        print(i)
        break
```

```
1
```

```
# 5. Find duplicates in an array
L = [1,1,2,3,4,4,5,5]
```

```
d = {}
```

```
for i in L:
    if i in d:
        d[i] = d[i] + 1
    else:
        d[i] = 1
```

```
for i in d:
    if d[i] > 1:
        print(i)
```

```
1
4
5
```

```
# 6. Rotate array to left d items
```

```
L = [1,2,3,4,5]
rotate = 2
```

```
for i in range(rotate):
    temp = L[0]
    for j in range(0, len(L)-1):
        L[j] = L[j+1]
    L[len(L)-1] = temp
```

```
print(L)
```

```
[3, 4, 5, 1, 2]
```

```
# 7. Find intersection of 2 sorted arrays
```

```
a = [1,2,3,4,5,8]
b = [3,6,7,8]
```

```
for i in a:
    if i in b:
        print(i)
```

```
3
8
```

```
a = [1,2,3,4,5,8]
b = [3,6,7,8]
```

```
i=j=0
```

```
while i<len(a) and j<len(b):
    if a[i] == b[j]:
        print(a[i])
        i+=1
        j+=1
    elif a[i] > b[j]:
        j+=1
    else:
        i+=1
```

```
3
8
```

```
# 8. Find continuous subarray with a given sum(given non-negative numbers)
# return the starting and ending index of the subarray
# return 1st subarray in case of multiple
```

```
L = [1,22,13,7,9,11,10]
S = 16
```

```
for i in range(0,len(L)):
    subarray = []
    for j in range(i,len(L)):
        subarray.append(L[j])
        if sum(subarray) == S:
            print(subarray)
```

```
[11, 10]
```

```
L = [1,22,13,7,9,11,10]
```

```
S = 35
```

```
d = {}
curr_sum = 0
```

```
for i in range(len(L)):
    curr_sum = curr_sum + L[i]

    if (curr_sum - S) in d:
        print(d[curr_sum - S]+1,i)
        break
```

```
d[curr_sum] = i
```

```
1 2
```

```
# 9. Find element with left side smaller/right side greater in an array
```

```
L = [3,1,2,5,8,7,9]
```

```
for i in range(1,len(L)-1):
    flag = True
```

```
    for j in range(0,i):
        if L[j] > L[i]:
            flag = False
```

```
    for k in range(i+1,len(L)):
        if L[k] < L[i]:
            flag = False
```

```
    if flag:
        print(L[i])
```

```
L = [3,1,2,5,8,7,9]
```

```
for i in range(1,len(L)-1):
    if max(L[:i]) < L[i] < min(L[i+1:]):
        print(L[i])
```

```
5
```

```

L = [3,1,2,5,8,7,9]

max_arr = []
min_arr = []

max_val = L[0]
min_val = L[-1]

for i in L:
    if i>max_val:
        max_val = i
    max_arr.append(max_val)

for i in range(len(L)-1,-1,-1):
    if L[i] < min_val:
        min_val = L[i]

    min_arr.insert(0,min_val)

for i in range(1,len(L)-1):
    if max_arr[i-1] < L[i] < min_arr[i+1]:
        print(L[i])

5

```

```

# 10. Maximum sum subarray
L = [-2,4,7,-1,6,-11,14,3,-1,-6]

d = {}

for i in range(0,len(L)):
    subarray = []
    for j in range(i,len(L)):
        subarray.append(L[j])
        d[sum(subarray)] = subarray

max_val = max(d.keys())

for i in d:
    if i == max_val:
        print(d[i])

[4, 7, -1, 6, -11, 14, 3, -1, -6]

```

```

L = [-2,4,7,-1,6,-11,14,3,-1,-6]

```

```

curr_sum = 0
curr_seq = []
best_sum = L[0]
best_seq = []

for i in L:
    if i + curr_sum > i:
        curr_sum = curr_sum + i
        curr_seq.append(i)
    else:
        curr_sum = i
        curr_seq.clear()
        curr_seq.append(i)

    if curr_sum > best_sum:
        best_sum = curr_sum
        best_seq = curr_seq

print(best_sum,best_seq)

```

```

22 [4, 7, -1, 6, -11, 14, 3, -1, -6]

```

```

-9223372036854775808

```

```

# 11. Sort arrays with items 1 and 0

```

```
# 12. Move all -ve numbers to the end
```

```
# 13. Maximum Product Subarray
```