

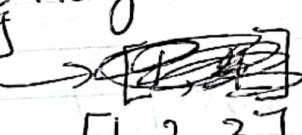
Y3 5.12 Sample Offline Data

21 minute to write pseudocode

and 41 min total

Constraints:

I/O: Array and size / subset of given size

- Do we just return a random subset?
- What should be the values inside the array?
- The problem says "distinct" elements, does this mean that all items in the array is unique?
- Does the order of returned array matter?
- Is the array input sorted?
- Can I use a python function `random()` to return a random subset?
 - Should I pick ~~any~~ items out randomly or should I ~~pick~~ return a ~~a~~ subset of the array of given ~~a~~ size.


size=3 [1, 2, 3, 4, 5] [1, 2, 3] or [1, 3, 5]?

What does it mean by random?

↳ There is no true random in programming

I can generate a random number less than or equal to the size input and
delete or create a new list and append from the input array

2/3 Input(A: List[int], s: int)

Pseudocode:

Let some variable be a random number
that is greater than or equal to 0 and
less than or equal to the input size

$r \in [0, s]$

$r = \text{random \#}$

Should I slice? or delete from list?

Use loop from 1 to $(\text{len}(A) - s)$

Ex [1, 2, 3, 4, 5] $s = 3$

Here, we want to delete from list two
times to get a return value of size three.

~~for i in range(1, len(A) - s):~~

for i in range(1, len(A) - s)

Generate r
Delete item at position r
del. A[r]

return A

3/3

def. fun(A: List[int], s: int) → List[int]:
 for i in range(1, len(A)-s):
 r = rand() # random
 del A[r]
 return A

Evaluation:

- My thought process was correct and similar to the book's sol'n
- Forgot to state time, space complexity $O(n), O(1)$
- Some parts were corrected when writing code
 - for i in range(1, len(A)-s)
should be
 - for i in range(0, len(A)-s)
- I had to look up the Python function `rand` to generate random number (int) between

How could I be better?

- The book did not delete items ~~in list~~ instead iterate through list ~~in~~ the input size number of times and generated a random integer from between i and $\text{len}(A)-1$ and switch from position
- Consider where I am starting the iteration

5.18

+ 1 hr

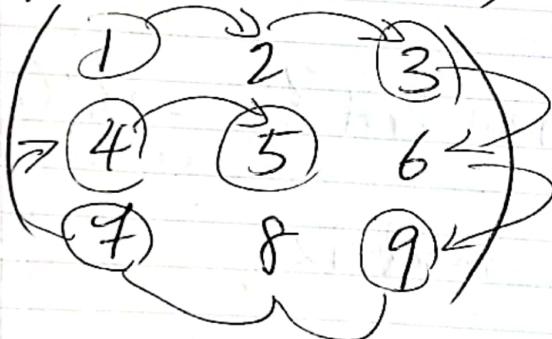
1/5

Input: $n \times n$ 2D array
Output: Array $\rightarrow [1, 2, 3, \dots]$ w/ spiral ordering

Hint says I use case analysis and divide and conquer
but I don't know neither.

What is the naive sol'n

Example: (from book)



Go right $n-1$ times
Go down $n-1$ times
Go left $n-1$ times
Go up $n-2$ times
Go right $(n-2)$ times

Instead, can I go from i to n increment, for every time I have to? Is there a simpler way than this?

I should use the coordinate

$[[1, 2, 3], [4, 5, 6], [7, 8, 9]]$

(nested list)

2/5

0 1 2

C₁ C₂ C₃

r₁ 0 [1, 2, 3],
r₂ 1 [4, 5, 6],
r₃ 2 [7, 8, 9]

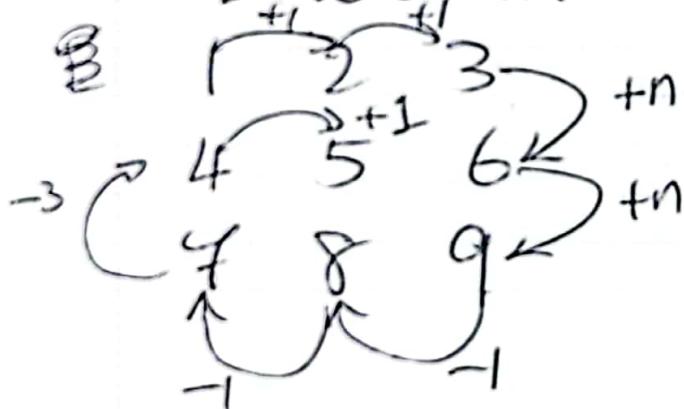
* can I make use of
fact that the matrix
is nxn?

for (r_i, c_j) in range Row, Column → O(n²)

What is the output?

↳ Should I make a new list? → O(n²)
~~= O(n)~~

When I look at the matrix,



Can I use this?

1 min

75.

0 1 2

C₁ C₂ C₃

1 0 [1, 2, 3],
2 1 [4, 5, 6],
3 2 [7, 8, 9]

* can I make use of
fact that the matrix
is $n \times n$?

for (r_i, c_i) in range Row, Column

What is the output?

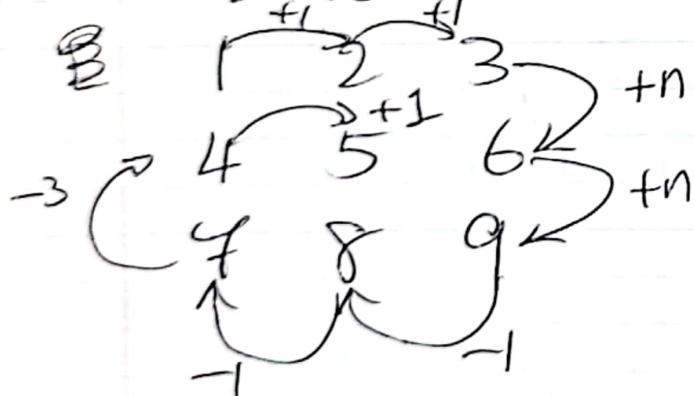
↳ Should I make a new list?

$\rightarrow O(n^2)$

$\rightarrow O(\text{constant})$

~~$O(n^2)$~~
 ~~$O(n)$~~
 ~~$O(1)$~~

When I look at the matrix,



Can I use this?

19 min

I want to try to loop once
for i in range(n)

$[A[0][0], A[0][1], A[0][2], \cancel{A[1][0]}, \cancel{A[1][1]}, \cancel{A[1][2]}]$

$A[1][2], A[2][2],$

$A[2][0], A[2][1],$

$A[1][0], \cancel{A[0][0]}, \cancel{A[0][1]}, \cancel{A[0][2]}$

$A[1][1]$

$res = [], x = 0$

for i in range(n):

$res.append(A[x][i])$

if $i == (n-1):$

$x = n-1$
 $res.append(A[i-1][x])$

if $i == (n-2):$

$res.append(A[x][i-1])$

if $i == 0:$

$res.append(A[0][0])$

4/5

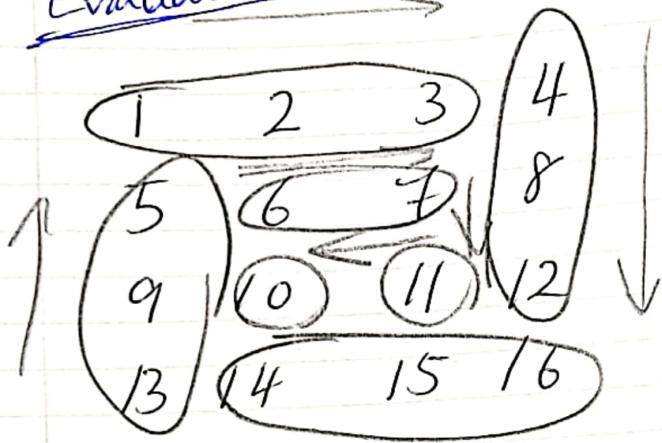
~~if~~
if $i == (n-2)$:

$x = n-2$

res.append($A[x][i+1]$)

33 min

Evaluation



Make a uniform way of adding boundary.

$n-1$ first row, $n-1$ last column, $n-1$ first column reversed.

My code was all over the place
I had the same thought process as
the solution although it was ambiguous
and I could not code it.

I need to take note on boundaries
and the constraints.

I took some time thinking on how I can
make this into $O(n)$ time. There is no sol'n
where I can achieve this and wasted too
much time.

See NeetCode sol'n

5.17) Sudoku Prob. Check

Input: 9×9 2D array; Output: Boolean

Want to check for every 3×3 subarray in the 9×9 , all the items inside are unique and $\in [1, 9]$.

\emptyset means that the entry is blank.

What will the input look like?

$[[\dots], [\dots], [\dots], \dots]$

So there are 9^2 total entries inside
use for edge case?

If entry $\notin [1, 9]$ return False?

Should I go row by row checking
if $\in [1, 9]$ or per 3×3 subarray? This prob only

Should I make boundary for
each 3×3 ?



If I make a loop for 3×3 , then

for loops

Maybe I should check for unique $\epsilon[1, 9]$ for every three subarray of the input.

Ex

$\left[[1, 2, 3], [4, 5, 6], [7, 8, 9] \right]$, $\underline{[1, 2, 3]} \cdot \underline{[4, 5, 6]}$,

restart check
again

check if unique $\epsilon[1, 9]$

In this case, I can use a for loop to check 3 arrays at a time.

I can use a hashmap to check uniqueness and $\epsilon[1, 9]$ if in the domain.

Then how can I move onto the next 3×3 ?

This will take ~~$O(q^2)$~~ or $O(n^2)$ time.
and take $O(n)$ space since hashmap

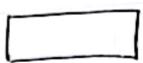


14 min

How can I check rows?

~~def fun~~

def fun(A: List[List[int]]) → Boolean
dict = {}
for ~~entry~~ in A:



For the ~~3x3~~ sudoku, can I just add up all the entries and check if it's below $1+2+\dots+9$? and ~~check~~ check for unique entry? ①

	0	1	2	3	4	5
0	A _{0,0}	A _{1,0}	A _{2,0}			
1						
2						
3						
4						
5						

if unique, add to total.

```

for x in range(3)
    (Check if unique), if yes, add to total
    for y in range(3)
        Total += A[x][y]
next_row=0
next_col=0
for row in range(3):
    for col in range(next_col+3): (next_col+3):
        total += A[row][col]
        if A[row][col] not in dict
            and A[row][col] ∈ {0, 9}:
            total += A[row][col]
next_row+=3
next_col+=3
There should be some kind of constraint
to check if in

```

28 min