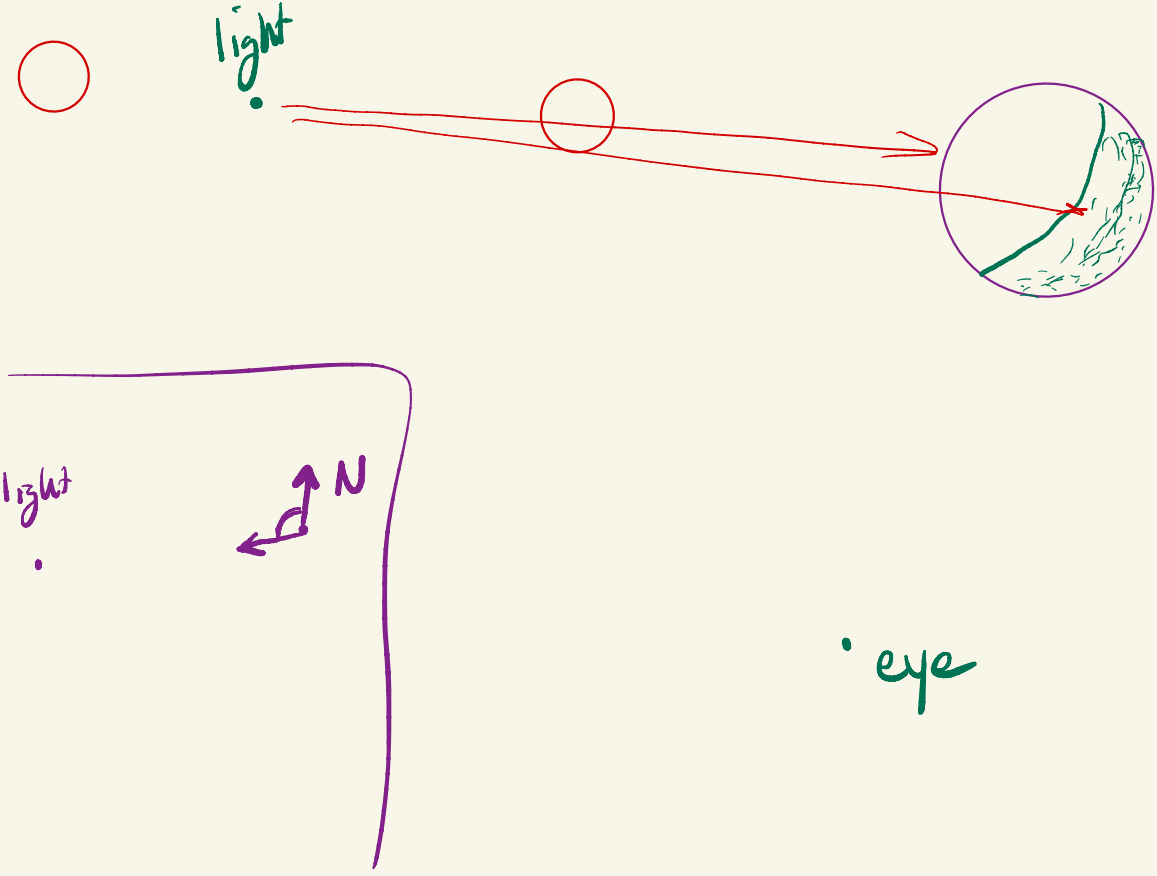
CPE/CSC 101

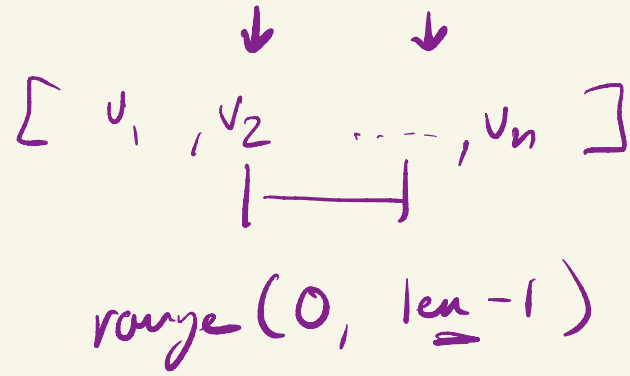
Today

- Selection Sort
- Files


light
light
N
eye
Ambient

Write a function that takes three arguments: a list of `int` values, a "lower index" `int` value, and an "upper index" `int` value. This function must return the smallest value in the input list between the lower index and the upper index (both bounds are inclusive). If there is no such value, the function must return `None`. If the either the lower index or the upper index is out of bounds, the function should return `None`.

(You may not use the min library function; that's not the goal here.)

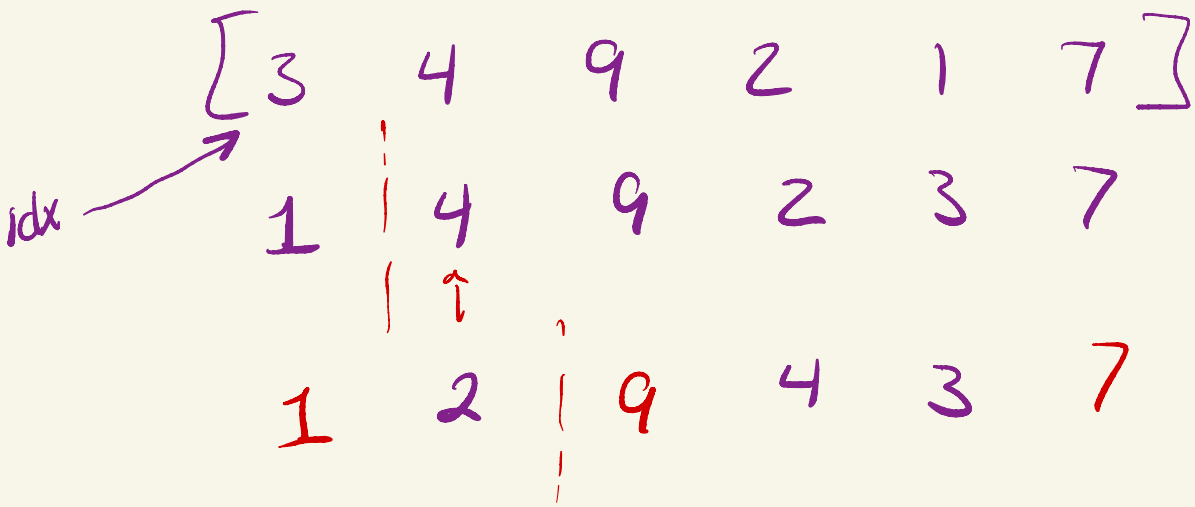



```
smallest = L[lower]
for idx in range(lower, upper+1)
        if L[idx] < smallest:
            smallest = L[idx]
return smallest
```

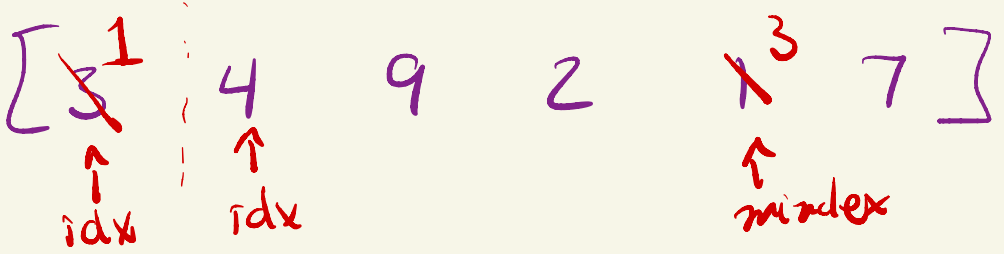
```
def small(L: List[int], lower: int, upper: int) -> int:
    if L == []:
        return None
    if not(0<= lower and lower < len(L)):
        return None
    if not(0<= upper < len(L)):
        return None

    if lower > upper:
        return None

    smallest = L[lower]
    for idx in range(lower, upper+1):
        if L[idx] < smallest:
            smallest = L[idx]
    return smallest
```

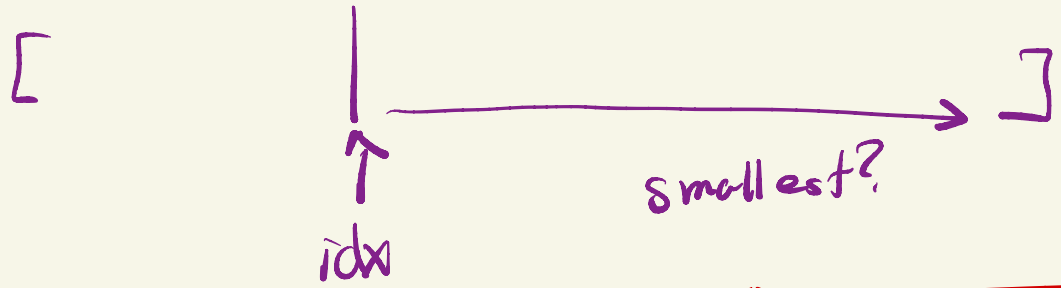
Optional[int]

Selection Sort



```
def selection_sort(L: List[int]) -> None:
    for idx in range(len(L)):
        mindex = smallest_from_index(L, idx)
        tmp = L[idx]
        L[idx] = L[mindex]
        L[mindex] = tmp
```





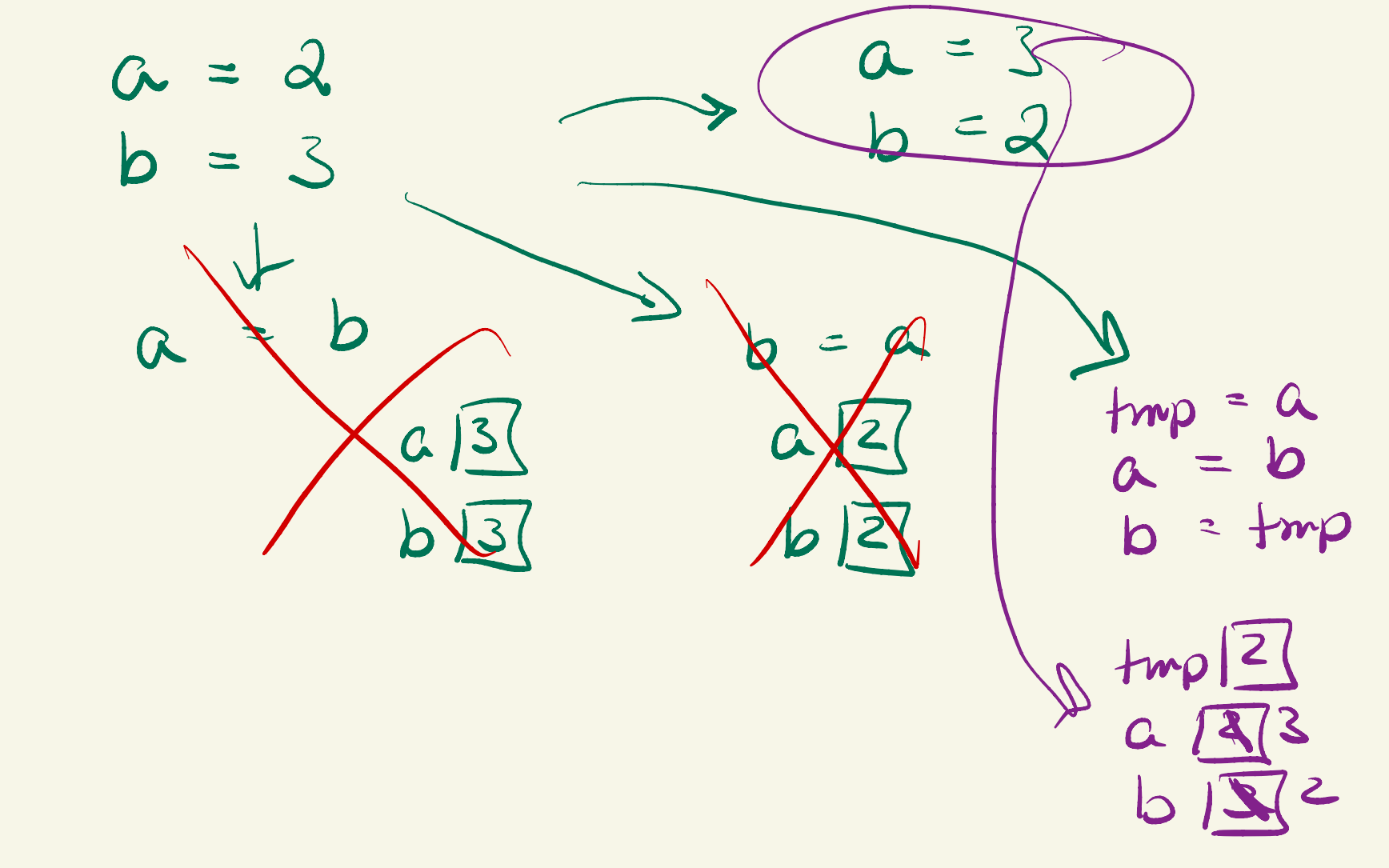
L = ...

selection_sort(L)

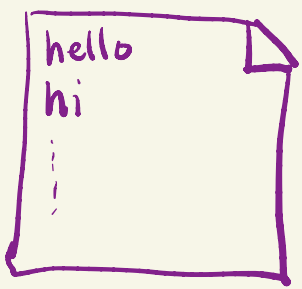
← L is sorted

n = length of L

n^2 operations


a = 2
b = 3
a = b
a 3
b 3
b = a
a 2
b 2
a = 3
b = 2
tmp = a
a = b
b = tmp
tmp 2
a 3
b 2

File



hello\nhi\n ...

newline

There can be errors - exceptions (files)

→ it doesn't exist

→ no permission