

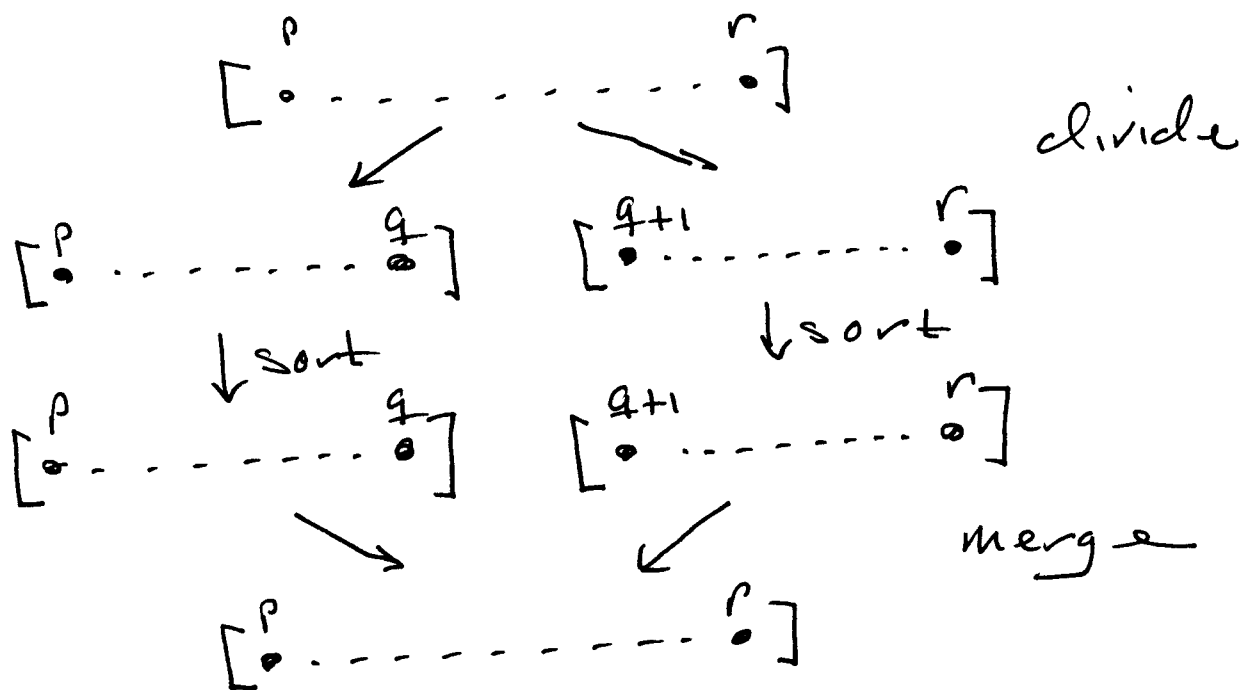
CS 230 4-6-21

Replacement Lecture

- Quiz 1 : today 6pm - 2am
- Pal : extended 1 more day
- lab1 : due mon 5/12

More Recursion

Ex. Sorting a list : Merge Sort

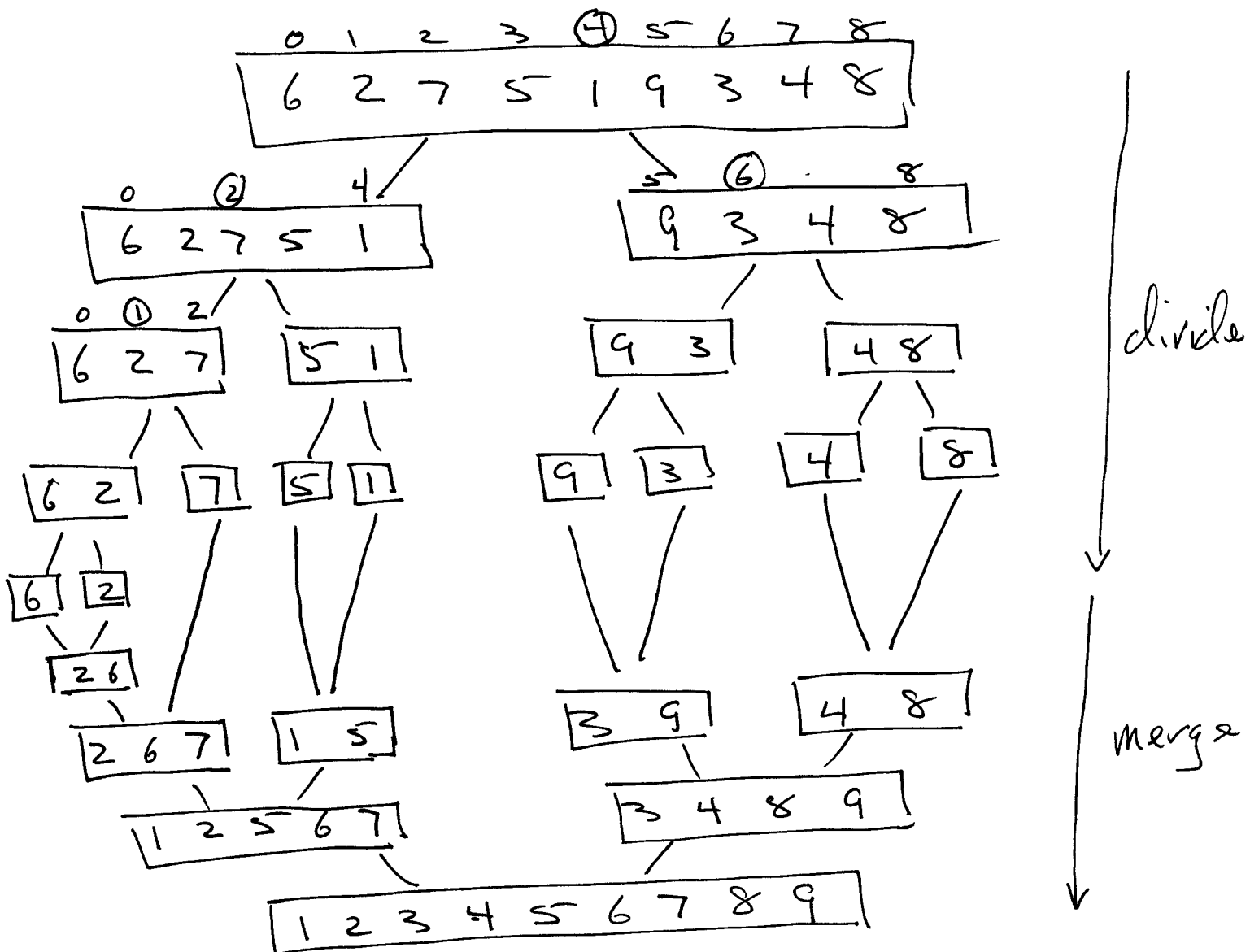


The recursion bottoms out when

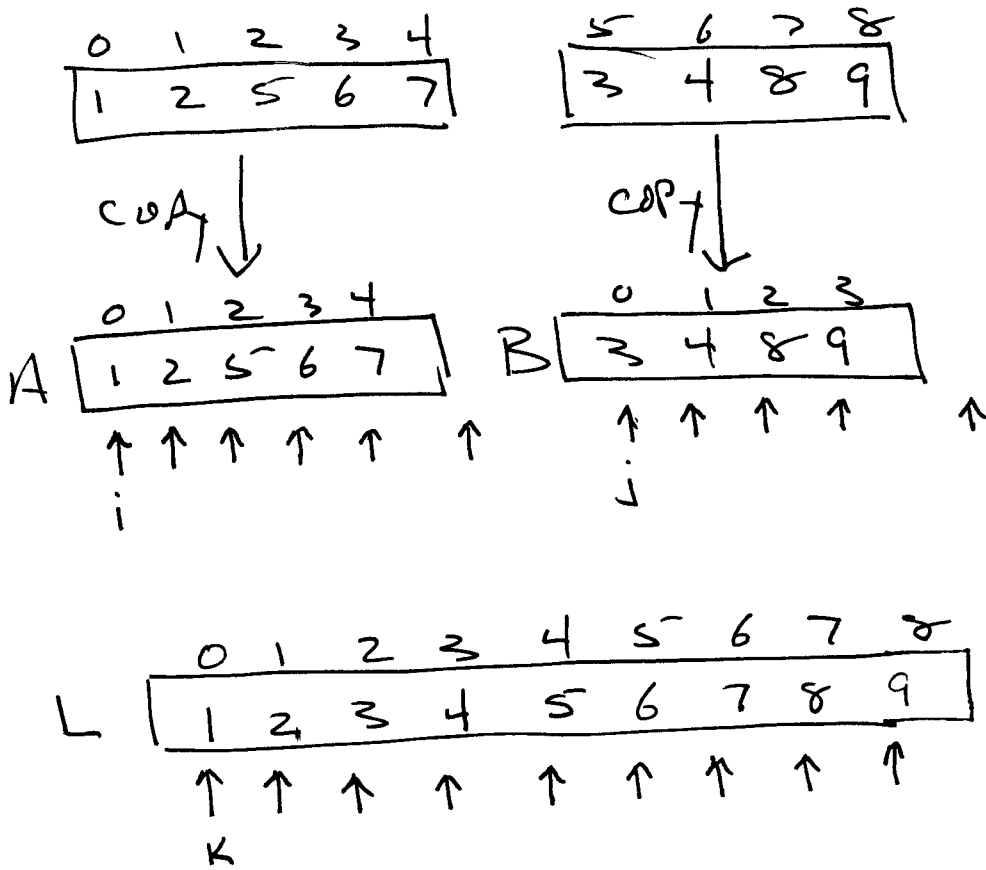
$P = r$, i.e. length of sublist is 1.

• see Sorting Searching, Py

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



how does merge work?



Problem: Search a list for a target x . return index where found (if found) or return None if not found.

binarySearch(L, p, r, x)

↑

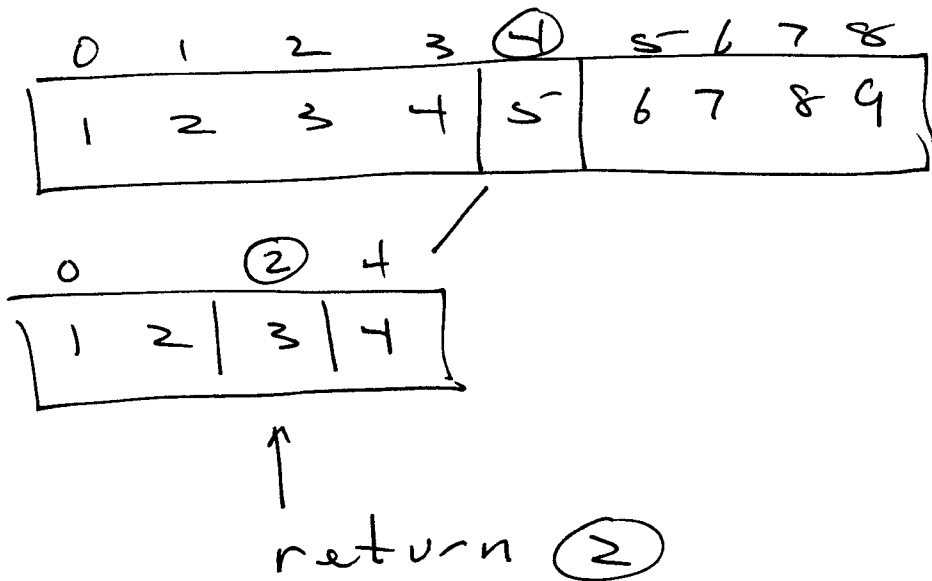
sub-list

↑

target

sorted L = [1, 2, 3, 4, 5, 6, 7, 8, 9]

Box trace: x = 3



Box trace: $x = 9$

0	1	2	3	4	5	6	7	8
1	2	3	4	5	6	7	8	9

5	6	7	8
6	7	8	9

7	8
8	9

8
9

↑ return 8

Box trace: 15

1	2	3	4	5	6	7	8	9
---	---	---	---	---	---	---	---	---

6	7	8	9
---	---	---	---

7	8
8	9

8
9

empty
↑

return None

Generators

first: iterables/iterators

A Python object is iterable if we can use it in a for loop

```
for x in iterable:
    print(x)
```

for instance: list, tuples, set, dictionaries
string, range object, enumerate obj,
... more ...

A class is iterable iff it implements methods

```
-- iter--() } iterable
-- next--() } protocol
```

more on classes later.

7

An iterable object can be used to create an iterator object by using built-in `for iter()`.

`L = [1, 2, 3]`

`I = iter(L)`

now

`for x in I:`

`print(x)`

can also iterate over `I` with built-in `for next()`

`I = iter(L)`

`next(I) # 1`

`next(I) # 2`

`next(I) # 3`

`next(I) # raises StopIteration exception.`

$I = \text{iter}(L)$

$\text{next}(I, 'foo') \neq 1$

$\text{next}(I,) \neq 2$

$\text{next}(I,) \neq 3$

$\text{next}(I, 'foo') \neq 'foo'$

↑
sentinel value

$\text{next}()$ is only defined on iterators
(not iterables)

$\text{next}(L) \neq$ raises `TypeError`