



### Problem 1

Given a list, get the squares of the elements in another list.

$l = [5, 1, 2, 3, 7]$

$res = [25, 1, 4, 9, 49]$

### Problem 2

Odd  $\rightarrow$  same  
Even  $\rightarrow$  square

Odd ✓

1, 3, 5, ...

$l = [5, 1, 2, 3, 7, 6]$

↓ ↓ ↓ ↓ ↓ ↓

Even

2, 4, 6, ...

$res = [5, 1, 4, 3, 7, 36]$

Odd  $i \% 2 \neq 0$

Even  $i \% 2 == 0$

---

set = { 1, 5, 5, 7, 2 }

dict = { k:v, k<sub>2</sub>:v<sub>2</sub>, ... }

---

`l = []`

`for i in range(3):`

~~`l.append(i)`~~

`for j in range(2):`

`l.append(i*j)`

`[i for i in range(3)]`

`[i*j for j in range(2) for i in range(3)]`

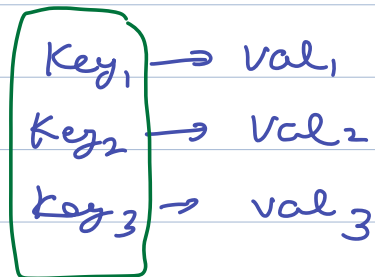
---

## Sets & Dicts

### ↳ Hashing

---

Dict



All of the keys are unique



Keys are restricted to be only immutable.  
Can't be mutable.

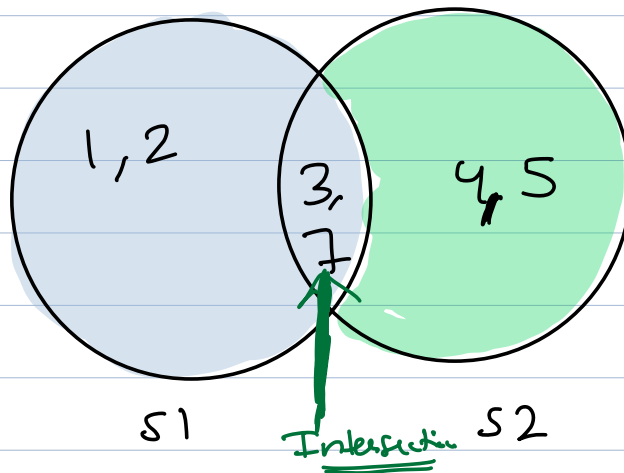
Values can be anything. ✓

---

Sets ⇒ all distinct values? Can't be mutable

↓  
Only immutable objects.

---



Union

$res = S1 \cup S2$

$res = S1.union(S2)$

$\{1, 2, 3, 4, 5, 7\}$

Intersection

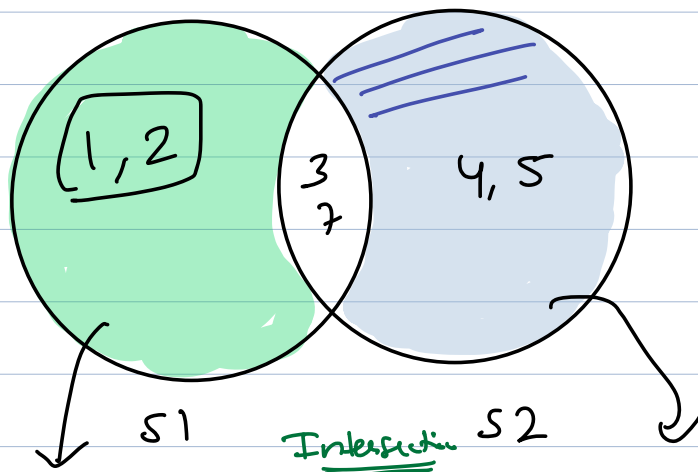
$res = S1 \& S2$

$res = S1.intersection(S2)$

$\{3, 7\}$

Difference

$res = S1 - S2$



$S1 - S2$

$S1.difference(S2)$

$\{1, 2\}$

$S2 - S1$

$S2.difference(S1)$

$\{4, 5\}$

## Symmetric Difference

→ present only once in the sets.

$$= (s1 \cup s2) - (s1 \cap s2)$$

res = s1. symmetric-difference(s2)

$$res = s1 \Delta s2$$

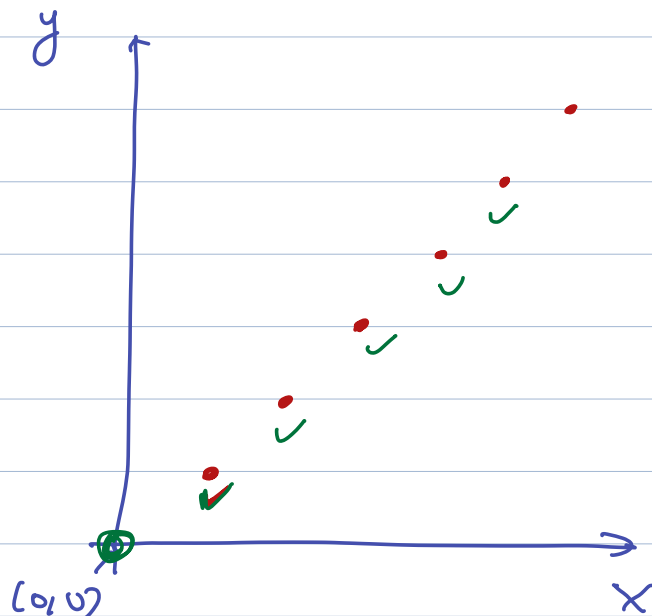
↑

{1, 2, 4, 5}

(xor sign)

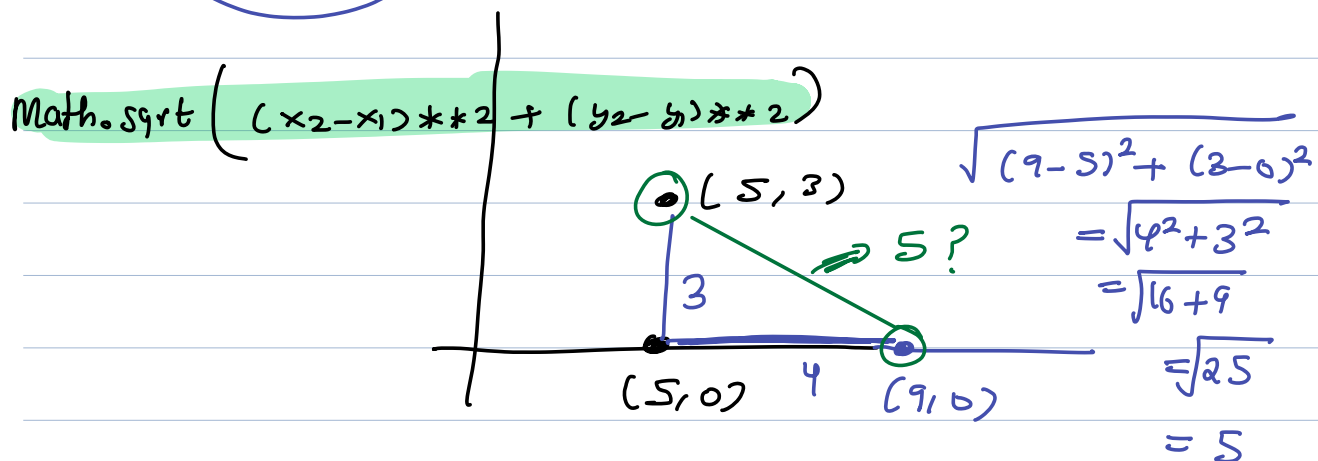
My:  $(0,0)$

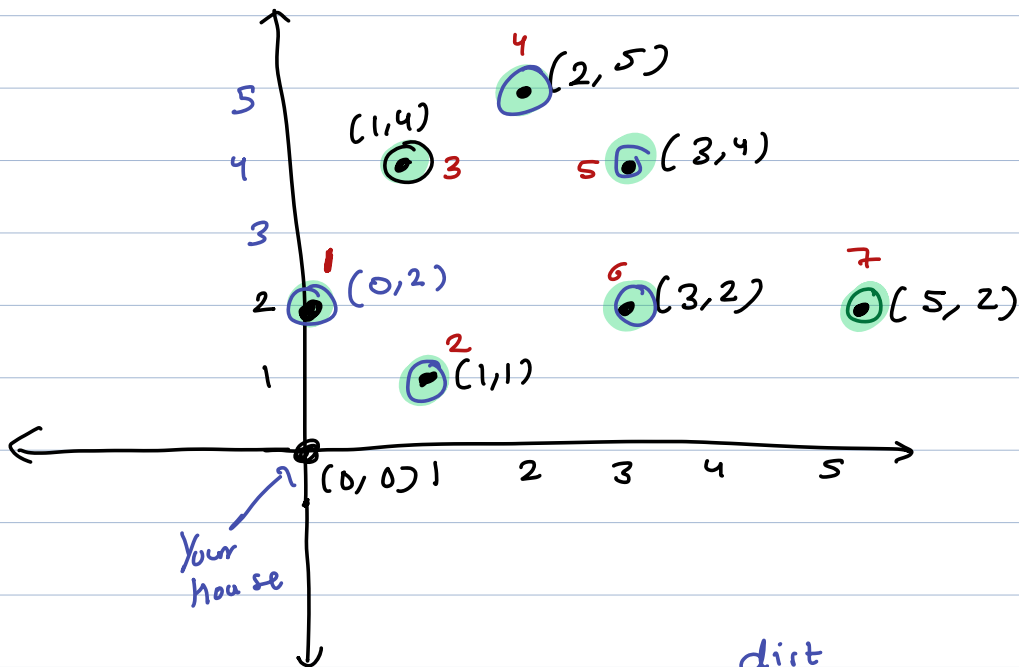
Tourist:  $[(1,1), (2,2), (3,3), (4,4), (5,5), (6,6)]$



Output =  $[(1,1), (2,2), (3,3), (4,4), (5,5)]$

$$\begin{matrix} (x_1, y_1) \\ \updownarrow \\ (x_2, y_2) \end{matrix} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$





✓ 1 - (0,2) — (0,0)

✓ 2 - (1,1) — (0,0)

✓ 3 - (1,4) — (0,0)

4 - (2,5) — (0,0)

✓ 5 - (3,4) — (0,0)

✓ 6 - (3,2) — (0,0)

7 - (5,2) — (0,0)

dist

$\boxed{2}$  ✓

$\sqrt{2} = 1.4$  ✓

$\sqrt{1+16} = \sqrt{17} = 4.123$  ✓

$\sqrt{4+25} = \sqrt{29} = 5.385$  ✓

$\boxed{5}$  ✓

$\sqrt{9+4} = \sqrt{13} = 3.606$  ✓

$\sqrt{25+4} = \sqrt{29} = 5.385$  ✓

output = [(1,1), (0,2), (3,2), (1,4), (3,4)]



Step-1 : Iterate over all points, calculate & store their distance from your place.

Step-2 : Sort the distances in ascending order.

Step-3 : Pick the top 5 corresponding points in the res.