

# [CS 11] Prac 2k – Zigzags

---

[oj.dcs.upd.edu.ph/problem/cs11prac2k](https://oj.dcs.upd.edu.ph/problem/cs11prac2k)

**Cheatsheet is available here:** <https://oj.dcs.upd.edu.ph/cs11cheatsheet/>

## Problem Statement

---

Three numbers form a zigzag if they either go up then down, or down and up. For example, the following form zigzag sequences:

- 3, 5, 4
- 3, 4, 3
- 3, 1, 3

The following do not:

- 3, 5, 7
- 6, 5, 2
- 5, 5, 5
- 3, 2, 2

Given a sequence of numbers, enumerate all of its length-3 subsequences that form a zigzag.

## Task Details

---

Implement a function called `zigzags`:

Copy

```
def zigzags(seq):
```

- `seq`—tuple of `ints`

Return a `frozenset` of triples of `ints`.

## Restrictions

---

(See 2a for more restrictions)

For this problem:

- Up to 11 function definition is allowed.

- Recursion is **disallowed**. (The recursion limit has been greatly reduced.)
- Comprehensions are allowed.
- `range` is allowed.
- The source code limit is 500500.

## Example Calls

---

### Example 1 Function Call

Copy

```
zigzags((3, 1, 4, 1, 5, 9, 2))
```

### Example 1 Return Value

Copy

```
frozenset()  
    (3, 1, 4), (3, 1, 5), (3, 1, 9), (3, 1, 2), (3, 4, 1), (3, 4, 2),  
    (3, 5, 2), (3, 9, 2), (1, 4, 1), (1, 4, 2), (1, 5, 2), (1, 9, 2),  
    (4, 1, 5), (4, 1, 9), (4, 1, 2), (4, 5, 2), (4, 9, 2), (1, 5, 2),  
    (1, 9, 2), (5, 9, 2),  
)
```

## Constraints

---

- The function `zigzags` will be called at most 100100 times.
- `seq` will have at most 88 elements.
- Each element of `seq` has absolute value at most  $102010^{20}$ .

## Scoring

---

- You get 100100 ❤ points if you solve all test cases.

[Report an issue](#)

## Clarifications

---

No clarifications have been made at this time.