Hash function

```
s = set()

s.add(8)

s

{8}

hash(8)

8

s.add("Rahul")

s

{8, 'Rahul'}

hash("Rahul")

-7732395848209247527

hash(11.1)

230584300921368587

hash(11.11)

253642731013505035


# That is why the searching, finding, deleting operations in set
# has TC -> 0(1)
```

Merging two list

```
left = [2, 8, 15, 18]
right = [5, 9, 12, 17]

def merging_lists(left, right):
    # Indexes of left and right lists
    i = 0 # Starting index of left list
    j = 0 # Starting index of right list
    res = []

    # Start comparing
    while i < len(left) and j < len(right):
        # do comparison
        if left[i] < right[j]:
            res.append(left[i])
```

```
            i += 1
        else:
            res.append(right[j])
            j += 1
        print(res)

    # If we are left with any elements in any of the lists
    if i < len(left):
        res += left[i:]
    else:
        res += right[j:]

    return res

left = [2, 8, 15, 18]
right = [5, 9, 12, 17, 19, 20, 22]

merging_lists(left, right)
```

```
[2]
[2, 5]
[2, 5, 8]
[2, 5, 8, 9]
[2, 5, 8, 9, 12]
[2, 5, 8, 9, 12, 15]
[2, 5, 8, 9, 12, 15, 17]
[2, 5, 8, 9, 12, 15, 17, 18]

[2, 5, 8, 9, 12, 15, 17, 18, 19, 20, 22]
```

```
# Get the half half sorted lists
# Merge these sorted lists
```

Merge sort code

```
arr = [9, 3, 7, 5, 6, 4, 8, 2]

def merge_sort(arr):
    # len of list/array
    n = len(arr)

    # base condition
    if n <= 1:
        return arr

    # Recursion calls on left and right halves
    left = merge_sort(arr[0:n//2])
    right = merge_sort(arr[n//2:])
```

```
    # Merging them

    i = 0
    j = 0

    res = []

    # Iterate on both lists then compare and merge
    while i < len(left) and j < len(right):
        if left[i] < right[j]:
            res.append(left[i])
            i += 1
        else:
            res.append(right[j])
            j += 1

    # If we are left with any ele in any list
    if i < len(left):
        res += left[i:]
    else:
        res += right[j:]

    return res

merge_sort(arr)
```

```
[2, 3, 4, 5, 6, 7, 8, 9]
```

Harmonic sum

```
# Get me round to 3 decimal places
```

Reverse a number