

HW 2

August 1, 2020

0.1 Merge sorted stream

```
[57]: def merge_sorted_stream(*args):  
    index = [iter(i) for i in args]  
    locatemin = [next(index[i]) for i in range(len(index))]  
    while locatemin:  
        min_index = locatemin.index(min(locatemin))  
        yield locatemin[min_index]  
        try:  
            locatemin[min_index] = next(index[min_index])  
        except StopIteration:  
            args = args[:min_index] + args[min_index+1:]  
            locatemin.pop(min_index)
```

```
[60]: stream1 = range(0, 10, 2)  
stream2 = range(1, 10, 2)  
for x in merge_sorted_stream(stream1, stream2):  
    print(x)
```

0
1
2
3
4
5
6
7
8
9

```
[63]: stream1 = range(0, 10, 2)  
stream2 = range(1, 10, 2)  
stream3 = [-1, 7, 7, 22]  
for x in merge_sorted_stream(stream1, stream2, stream3):  
    print(x)
```

-1
0
1
2
3
4
5
6
7
7
7
8
9
22

0.2 Tree traversal

```
[68]: class TreeNode:

    def __init__(self, val):
        self.val = str(val)
        self.left = None
        self.right = None

    def in_order(self):
        # Left, Root, Right
        if self.left:
            for order in self.left.in_order():
                yield order

        yield self.val

        if self.right:
            for order in self.right.in_order():
                yield order

    def pre_order(self):
        # Root, Left, Right
        yield self.val

        if self.left:
            for order in self.left.pre_order():
                yield order

        if self.right:
            for order in self.right.pre_order():
```

```

        yield order

def post_order(self):
    # Left, Right, Root
    if self.left:
        for order in self.left.post_order():
            yield order

    if self.right:
        for order in self.right.post_order():
            yield order

    yield self.val

root = TreeNode(1)
root.left = TreeNode(2)
root.right = TreeNode(3)
root.left.left = TreeNode(4)
root.left.right = TreeNode(5)

print(' -> '.join(item for item in root.in_order()))
print(' -> '.join(item for item in root.pre_order()))
print(' -> '.join(item for item in root.post_order()))

```

```

4 -> 2 -> 5 -> 1 -> 3
1 -> 2 -> 4 -> 5 -> 3
4 -> 5 -> 2 -> 3 -> 1

```