DAY-17

Min & Max Heap Implementations.

main Heap Implementation:

```
class MinHeap:
    def __init__(self):
        self.heap_list = [0]
        self.current_size = 0
    def sift_up(self, i):
        while i//2 > 0:
            if self.heap_list[i] < self.heap_list[i//2]:
                self.heap_list[i], self.heap_list[i//2]
                = self.heap_list[i//2], self.heap_list[i]
    # the above were swapped.
    def insert(self,k):
        self.heap_list.append(k)
        self.current_size += 1
        self.sift_up(self.current_size)

    def sift_down(self,i)
        while (i*2) <= self.current_size:
            mc = self.min_child(i)
            if self.heap_list[i] > self.heap_list[mc]:
                self.heap_list[i], self.heap_list[mc]
                = self.heap_list[mc], self.heap_list[i]
```

```
def min_child(self, i):
    if (i*2)+1 > self.current_size:
        return i*2
    else:
        if self.heap_list[i*2] < self.heap_list[(i*2)+1]:
            return i*2
        else:
            return (i*2)+1
def delete_min(self):
    if len(self.heap_list)==1:
        return "Empty Heap"
    root = self.heap_list[1]
    self.heap_list[1] = self.heap_list[self.current_size]
                                       current.size.

    *self.heap_list, _ = self.heap_list
    # last value popped.

    self.current_size -= 1
    self.sift_down(1).

    return root.
```