DSC 190 - Discussion 04

Problem 1.

When performing a search for the k nearest neighbors to a query point, we need to keep track of the k smallest distances found so far. We can do so using a heap.

Fill in the class below so that it keeps track of the k smallest numbers inserted while maintaining a heap whose size is never larger than $k + 1$.

```
class KSmallest:

    def __init__(self, k):
        ...

    def insert(self, number):
        """Insert a number."""
        ...

    def max(self):
        """Return the largest of the k numbers stored."""

    def as_list(self):
        """Return the k elements as a list."""
        ...
```

Solution:

```
class KSmallest:

    def __init__(self, k):
        self.k = k
        self.heap = MaxHeap()

    def insert(self, key):
        if len(self.heap.keys) < self.k or key < self.heap.max():
            self.heap.insert(key)

        if len(self.heap.keys) > self.k:
            self.heap.pop_max()

    def as_list(self):
        return list(self.heap.keys)

    def max(self):
        return self.heap.max()
```