Challenge 2: Find k smallest elements in a List

Given a list and a number "k" write a function that returns the first "k" smallest elements using a Heap.

We'll cover the following

Problem Statement#

Implement a function `findKSmallest(lst,k)` that takes an unsorted integer list as input and returns the "k" smallest elements in the list using a Heap. The `minHeap` class that was written in a previous lesson is prepended to this exercise so feel free to use it! Have a look at the illustration given for a clearer picture of the problem.

Output#

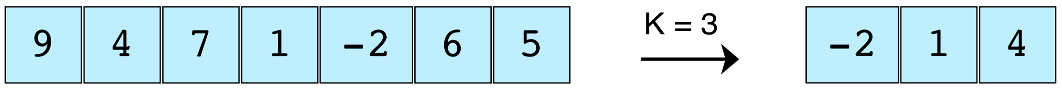
Returns integer list that contains the first k smallest elements from the given list.

Sample Input#

```
lst = [9,4,7,1,-2,6,5]
k = 3
```

Sample Output#

```
[-2,1,4]
```



Coding Exercise

Take a close look and design a step-by-step algorithm first before jumping onto the implementation. This problem is designed for your practice, so try to solve it on your own first. If you get stuck, you can always refer to the solution provided in the solution section. Good Luck!

main.py

MinHeap.py

```
class MinHeap:
    def __init__(self):
        self.heap = []

    def insert(self, val):
        self.heap.append(val)
        self.__percolateUp(len(self.heap)-1)

    def getMin(self):
        if self.heap:
            return self.heap[0]
        return None

    def removeMin(self):
        if len(self.heap) > 1:
            min = self.heap[0]
            self.heap[0] = self.heap[-1]
            del self.heap[-1]
            self.__minHeapify(0)
            return min
        elif len(self.heap) == 1:
            min = self.heap[0]
            del self.heap[0]
            return min
        else:
            return None

    def __percolateUp(self, index):
        parent = (index-1)//2
        if index <= 0:
            return
        elif self.heap[parent] > self.heap[index]:
            tmp = self.heap[parent]
            self.heap[parent] = self.heap[index]
            self.heap[index] = tmp
            self.__percolateUp(parent)

    def __minHeapify(self, index):
        left = (index * 2) + 1
        right = (index * 2) + 2
        smallest = index
        if len(self.heap) > left and self.heap[smallest] > self.heap[left]:
            smallest = left
        if len(self.heap) > right and self.heap[smallest] > self.heap[right]:
            smallest = right
        if smallest != index:
            tmp = self.heap[smallest]
            self.heap[smallest] = self.heap[index]
            self.heap[index] = tmp
            self.__minHeapify(smallest)

    def buildHeap(self, arr):
        self.heap = arr
        for i in range(len(arr)-1, -1, -1):
            self.__minHeapify(i)
```

← Back

Solution Review: Convert Max-Heap t...

Next →

Solution Review: Find k smallest elem...

☑ Mark as Completed

Report an Issue