

# Min Heap (Implementation)


We'll implement a min heap in this lesson!

We'll cover the following ^

- Implementation
- Explanation

## Implementation #

Now that we have discussed all the functions of a Min-Heap, we've implemented them in the following code executable. Run and test the code on multiple outputs to see if it returns the elements in the correct order every time? Try it!

 MinHeap.py

```
1 class MinHeap:
2     def __init__(self):
3         self.heap = []
4
5     def insert(self, val):
6         self.heap.append(val)
7         self.__percolateUp(len(self.heap)-1)
8
9     def getMin(self):
10        if self.heap:
11            return self.heap[0]
12        return None
13
14    def removeMin(self):
15        if len(self.heap) > 1:
16            min = self.heap[0]
17            self.heap[0] = self.heap[-1]
18            del self.heap[-1]
19            self.__minHeapify(0)
20            return min
21        elif len(self.heap == 1):
22            min = self.heap[0]
23            del self.heap[0]
24            return min
25        else:
26            return None
27
28    def __percolateUp(self, index):
29        parent = (index-1)//2
30        if index <= 0:
31            return
32        elif self.heap[parent] > self.heap[index]:
33            tmp = self.heap[parent]
34            self.heap[parent] = self.heap[index]
35            self.heap[index] = tmp
36            self.__percolateUp(parent)
37
38    def minHeapify(self, index):
```

```

39     left = (index * 2) + 1
40     right = (index * 2) + 2
41     smallest = index
42     if len(self.heap) > left and self.heap[smallest] > self.heap[left]:
43         smallest = left
44     if len(self.heap) > right and self.heap[smallest] > self.heap[right]:
45         smallest = right
46     if smallest != index:
47         tmp = self.heap[smallest]
48         self.heap[smallest] = self.heap[index]
49         self.heap[index] = tmp
50         self.__minHeapify(smallest)
51
52     def buildHeap(self, arr):
53         self.heap = arr
54         for i in range(len(arr)-1, -1, -1):
55             self.__minHeapify(i)
56
57
58     heap = MinHeap()
59     heap.insert(12)
60     heap.insert(10)
61     heap.insert(-10)
62     heap.insert(100)
63
64     print(heap.getMin())
65     print(heap.removeMin())
66     print(heap.getMin())
67     heap.insert(-100)
68     print(heap.getMin())
69

```



×

Output

0.155s

```

-10
-10
10
-100

```

## Explanation #

The code above for min heaps is an exact reflection of the code for max heaps! It's a good exercise to try and figure out what changed. However, if you have any confusion about it, leave a question on the community forum and we'll get back to you!

And now that we have covered both the implementations, let's try to solve some practice questions using the Heap data structure in the next few lessons!

← Back

Next →

 **Mark as Completed**



Report an  
Issue



Ask a Question

([https://discuss.educative.io/tag/min-heap-implementation\\_\\_introduction-to-heap\\_\\_data-structures-for-coding-interviews-in-python](https://discuss.educative.io/tag/min-heap-implementation__introduction-to-heap__data-structures-for-coding-interviews-in-python))