

Prompt	Scratchpad	Our Solution(s)	Video Explanation	Run Code	Your Solutions	Run Code
--------	------------	-----------------	-------------------	----------	----------------	----------

Solution 1	Solution 2	Solution 3
<pre>1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved. 2 3 using System.Collections.Generic; 4 5 public class Program { 6 public class MinHeap { 7 public List<int> heap = new List<int>(); 8 9 public MinHeap(List<int> array) { 10 heap = buildHeap(array); 11 } 12 13 // O(n) time O(1) space 14 public List<int> buildHeap(List<int> array) { 15 int firstParentIdx = (array.Count - 2) / 2; 16 for (int currentIndex = firstParentIdx; currentIndex >= 0; currentIndex--) { 17 siftDown(currentIdx, array.Count - 1, array); 18 } 19 return array; 20 } 21 22 // O(log(n)) time O(1) space 23 public void siftDown(int currentIndex, int endIdx, List<int> heap) { 24 int childOneIdx = currentIndex * 2 + 1; 25 while (childOneIdx <= endIdx) { 26 int childTwoIdx = currentIndex * 2 + 2 <= 27 endIdx ? currentIndex * 2 + 2 : -1; 28 int idxToSwap; 29 if (childTwoIdx != -1 && heap[childTwoIdx] < heap[childOneIdx]) { 30 idxToSwap = childTwoIdx; 31 } else { 32 idxToSwap = childOneIdx; 33 } 34 if (heap[idxToSwap] < heap[currentIdx]) { 35 swap(currentIdx, idxToSwap, heap); 36 currentIndex = idxToSwap; 37 childOneIdx = currentIndex * 2 + 1; 38 } else { 39 return; 40 } 41 } 42 } 43 44 // O(log(n)) time O(1) space 45 public void siftUp(int currentIndex, List<int> heap) { 46 int parentIdx = (currentIndex - 1) / 2; 47 while (currentIndex > 0 && heap[currentIdx] < heap[parentIdx]) { 48 swap(currentIdx, parentIdx, heap); 49 currentIndex = parentIdx; 50 parentIdx = (currentIndex - 1) / 2; 51 } 52 } 53 54 public int Peek() { 55 return heap[0]; 56 } 57 58 public int Remove() { 59 swap(0, heap.Count - 1, heap); 60 int valueToRemove = heap[heap.Count - 1]; 61 heap.RemoveAt(heap.Count - 1); 62 siftDown(0, heap.Count - 1, heap); 63 return valueToRemove; 64 } 65 66 public void Insert(int value) { 67 heap.Add(value); 68 siftUp(heap.Count - 1, heap); 69 } 70 71 public void swap(int i, int j, List<int> heap) { 72 int temp = heap[j]; 73 heap[j] = heap[i]; 74 heap[i] = temp; 75 } 76 } 77 } 78</pre>	<pre>1 using System.Collections.Generic; 2 3 // Do not edit the class below except for the buildHeap, 4 // siftDown, siftUp, Peek, Remove, and Insert methods. 5 // Feel free to add new properties and methods to the class. 6 public class Program { 7 public class MinHeap { 8 public List<int> heap = new List<int>(); 9 10 public MinHeap(List<int> array) { 11 heap = buildHeap(array); 12 } 13 14 public List<int> buildHeap(List<int> array) { 15 // Write your code here. 16 return null; 17 } 18 19 public void siftDown(int currentIndex, int endIdx, List<int> heap) { 20 // Write your code here. 21 } 22 23 public void siftUp(int currentIndex, List<int> heap) { 24 // Write your code here. 25 } 26 27 public int Peek() { 28 // Write your code here. 29 return -1; 30 } 31 32 public int Remove() { 33 // Write your code here. 34 return -1; 35 } 36 37 public void Insert(int value) { 38 // Write your code here. 39 } 40 } 41 } 42</pre>	
<div>Custom OutputRaw OutputSubmit Code</div>		

Run or submit code when you're ready.