Our Solution(s) Run Code

Solution 1

```cpp
// Copyright © 2020 AlgoExpert, LLC. All rights reserved.

#include <vector>
using namespace std;

class MinHeap {
public:
  vector<int> heap;

  MinHeap(vector<int> vector) { heap = buildHeap(&vector); }

  // O(n) time | O(1) space
  vector<int> buildHeap(vector<int> *vector) {
    int firstParentIdx = (vector->size() - 2) / 2;
    for (int currentIdx = firstParentIdx; currentIdx >= 0; curre
      siftDown(currentIdx, vector->size() - 1, vector);
    }
    return *vector;
  }

  // O(log(n)) time | O(1) space
  void siftDown(int currentIdx, int endIdx, vector<int> *heap) {
    int childOneIdx = currentIdx * 2 + 1;
    while (childOneIdx <= endIdx) {
      int childTwoIdx = currentIdx * 2 + 2 <= endIdx ? currentId
      int idxToSwap;
      if (childTwoIdx != -1 && heap->at(childTwoIdx) < heap->at(
        idxToSwap = childTwoIdx;
      } else {
        idxToSwap = childOneIdx;
      }
      if (heap->at(idxToSwap) < heap->at(currentIdx)) {
        swap(currentIdx, idxToSwap, heap);
```

Your Solutions Run Code

Solution 1 Solution 2 Solution 3

```cpp
#include <vector>
using namespace std;

// Do not edit the class below except for the buildHeap,
// siftDown, siftUp, peek, remove, and insert methods.
// Feel free to add new properties and methods to the class.
class MinHeap {
public:
  vector<int> heap;

  MinHeap(vector<int> vector) { heap = buildHeap(&vector); }

  vector<int> buildHeap(vector<int> *vector) {
    // Write your code here.
    return {};
  }

  void siftDown(int currentIdx, int endIdx, vector<int> *heap) {
    // Write your code here.
  }

  void siftUp(int currentIdx, vector<int> *heap) {
    // Write your code here.
  }

  int peek() {
    // Write your code here.
    return -1;
  }

  int remove() {
    // Write your code here.
    return -1;
```

Our Tests

Custom Output Submit Code

Run or submit code when you're ready.