AlgoExpert Quad Layout C++ 12px Sublime Monokai 00:00:00

Prompt Scratchpad Our Solution(s) Video Explanation Run Code

Solution 1

```cpp
    }
  }

  void siftUp(int currentIdx, vector<int> *heap) {
    int parentIdx = (currentIdx - 1) / 2;
    while (currentIdx > 0) {
      if (comparisonFunc(heap->at(currentIdx), heap->at(parentIdx))) {
        swap(currentIdx, parentIdx, heap);
        currentIdx = parentIdx;
        parentIdx = (currentIdx - 1) / 2;
      } else {
        return;
      }
    }
  }

  int peek() { return heap[0]; }

  int remove() {
    swap(0, heap.size() - 1, &heap);
    int valueToRemove = heap.back();
    heap.pop_back();
    length--;
    siftDown(0, heap.size() - 1, &heap);
    return valueToRemove;
  }

  void insert(int value) {
    heap.push_back(value);
    length++;
    siftUp(heap.size() - 1, &heap);
  }

  void swap(int i, int j, vector<int> *heap) {
    int temp = heap->at(j);
    heap->at(j) = heap->at(i);
    heap->at(i) = temp;
  }
};

class ContinuousMedianHandler {
public:
  Heap lowers;
  Heap greaters;
  double median;

  ContinuousMedianHandler()
      : lowers(MAX_HEAP_FUNC, {}), greaters(MIN_HEAP_FUNC, {}) {
    median = 0;
  }

  // O(log(n)) time | O(n) space
  void insert(int number) {
    if (!lowers.length || number < lowers.peek()) {
      lowers.insert(number);
    } else {
      greaters.insert(number);
    }
    rebalanceHeaps();
    updateMedian();
  }

  void rebalanceHeaps() {
    if (lowers.length - greaters.length == 2) {
      greaters.insert(lowers.remove());
    } else if (greaters.length - lowers.length == 2) {
      lowers.insert(greaters.remove());
    }
  }

  void updateMedian() {
    if (lowers.length == greaters.length) {
      median = ((double)lowers.peek() + (double)greaters.peek()) / 2;
    } else if (lowers.length > greaters.length) {
      median = lowers.peek();
    } else {
      median = greaters.peek();
    }
  }

  double getMedian() { return median; }
};

bool MAX_HEAP_FUNC(int a, int b) { return a > b; }

bool MIN_HEAP_FUNC(int a, int b) { return a < b; }
```