Q1: give mathematical formula to calculate the height of a tree with n nodes.
Solution:
floor(log2(n)) + 1
Q2: write C++ code for heap sort.
Solution:

```
void heapify(int arr[], int n, int i) {
  // Largest element index
  int largest = i;
  int left = 2 * i + 1; // left = 2*i + 1
  int right = 2 * i + 2; // right = 2*i + 2

  // Check if left child is larger than root
  if (left < n && arr[left] > arr[largest])
    largest = left;

  // Check if right child is larger than largest so far
  if (right < n && arr[right] > arr[largest])
    largest = right;

  // If largest is not root
  if (largest != i) {
    std::swap(arr[i], arr[largest]);

    // Recursively heapify the affected sub-tree
    heapify(arr, n, largest);
  }
}

void heapSort(int arr[], int n) {
  // Build a max heap (rearrange array for descending order)
  for (int i = n / 2 - 1; i >= 0; i--)
    heapify(arr, n, i);

  // One by one extract an element from heap
  for (int i = n - 1; i > 0; i--) {
    // Move current root to end
    std::swap(arr[0], arr[i]);

    // Call max heapify on the reduced heap
    heapify(arr, i, 0);
  }
}
```