Consider a max-priority queue $Q$ implemented using a binary max-heap. We would like to design an `ExtractSecondLargest(Q)` operation, which returns the second largest key in $Q$ and deletes it from $Q$. The worst-case running time of this operation must be in $O(\log n)$. We assume that all keys in $Q$ are distinct integers.

(a) Write the pseudo-code of your `ExtractSecondLargest(Q)` algorithm. Let $Q$ be an array whose indices start from 1. You can use operations from the textbook and lectures as helpers without describing their details.

(b) Explain why your pseudo-code works correctly. In particular, explain why the element you extract is the second largest one, and why this operation maintains the heap property.

(c) Explain why the worst-case running time of your algorithm is in $O(\log n)$.