Name: Alikhan Nurzhan
Group: SE-2429

Asymptotic complexity

Max Heap is a non-linear data structure, one of which's main strengths is the ability the get the largest element in O(1) time. Max Heap is implemented using PriorityQueue and has these complexities:

a) Insert : Θ(log n), O(log n), Ω(log n). Space: O(1)

b) get max : Θ(1), O(1), Ω(1). Space: O(1)

c) extract max: Θ(log n), O(log n), Ω(log n). Space: O(1)

d) increase key: Θ(n), O(2n+logn) due to 'contains' and 'remove', Ω(1). Space: O(1)

No recursion was used.

Code Review

Max Heap was implemented by using builtin java functionality of PriorityQueue, thus I cannot judge how effective was the sorting and swapping in the heap.

Usage of 'contains' and 'remove' in 'increase key' forces machine to go through the whole heap twice, worst case scenario being O(2n). In addition, 'increase key' becomes a lot more unusable when the key to be increased has duplicates.

There is also a distinct lack of ways to find nodes though their index, making testing the increase key tricky.

Memory usage is efficient and code is written to be easily readable and understandable.