10017168

Assignment 2 CPSC 331

Josh Schijns

Question 1

a) An algorithm would be~~a~~

~~lg~~ largest_Number = A[n-1-K]

b)
```
int largestNumber(int[] A, int low, int high)
    if low > high then
        throw notFoundException
    else
        mid = ⌊(low+high)/2⌋
        if (A[mid] > high) then
            return largestNumber(A, mid, high-1)
        elseif (A[mid] < high) then
            return largestNumber(A, low, mid-1)
        else
            return A[mid]
        end
    end
end
```

Question 2

```
int[] DPartition(int[] A, int low, int high)
    p = 0
    i = low
    j = high
    while i ≤ j do
        while i ≤ j and A[i] ≤ p do
            i = i+1
        while j ≥ i and A[j] ≥ p do
            j = j-1
        if i < j then
            Swap(A[i], A[j])
    return A
```

This will return an array that has all the negative ~~on~~ #'s ~~the~~ preceding the non-negative. The asymptotic run time is O(high-low)

$\therefore \Theta(n)$ ∎

Scanned by CamScanner

```
3. size = # of elements in stack
   while (size > 0) do
       max = Integer.Min_Value;
       For (int i=0, i<size, i++) do
           inttemp = stack.pop();
           if (temp > max) max = temp
           queue.add(temp);
       end
       for (int i=0; i<size; i++)
           temp = queue.remove()
           if ( temp == max) ~~stacktemp~~ do
               Stack.push(temp);
               i = size;
           else queue.add(temp);
       end
       size --;
   end
```

The algorithm is correct because ~~e~~ at the end of each loop in the while loop, the size decreases by 1, thus the loop guard will end once size=0, which is when the queue is empty and the stack is full and sorted. The max integer value is found and that value is pushed onto the stack. At the end the postconditions are met and the loop terminates. Its running time is $O(n^2)$.